

# HIGH PERFORMANCE NETWORKING

Edited by  
**Harmen van As**



IFIP



Springer  
Science+  
Business  
Media, LLC



---

# **HIGH PERFORMANCE NETWORKING**



## **IFIP – The International Federation for Information Processing**

IFIP was founded in 1960 under the auspices of UNESCO, following the First World Computer Congress held in Paris the previous year. An umbrella organization for societies working in information processing, IFIP's aim is two-fold: to support information processing within its member countries and to encourage technology transfer to developing nations. As its mission statement clearly states,

IFIP's mission is to be the leading, truly international, apolitical organization which encourages and assists in the development, exploitation and application of information technology for the benefit of all people.

IFIP is a non-profitmaking organization, run almost solely by 2500 volunteers. It operates through a number of technical committees, which organize events and publications. IFIP's events range from an international congress to local seminars, but the most important are:

- the IFIP World Computer Congress, held every second year;
- open conferences;
- working conferences.

The flagship event is the IFIP World Computer Congress, at which both invited and contributed papers are presented. Contributed papers are rigorously refereed and the rejection rate is high.

As with the Congress, participation in the open conferences is open to all and papers may be invited or submitted. Again, submitted papers are stringently refereed.

The working conferences are structured differently. They are usually run by a working group and attendance is small and by invitation only. Their purpose is to create an atmosphere conducive to innovation and development. Refereeing is less rigorous and papers are subjected to extensive group discussion.

Publications arising from IFIP events vary. The papers presented at the IFIP World Computer Congress and at open conferences are published as conference proceedings, while the results of the working conferences are often published as collections of selected and edited papers.

Any national society whose primary activity is in information may apply to become a full member of IFIP, although full membership is restricted to one society per country. Full members are entitled to vote at the annual General Assembly, National societies preferring a less committed involvement may apply for associate or corresponding membership. Associate members enjoy the same benefits as full members, but without voting rights. Corresponding members are not represented in IFIP bodies. Affiliated membership is open to non-national societies, and individual and honorary membership schemes are also offered.

---

# HIGH PERFORMANCE NETWORKING

IFIP TC-6 Eighth International Conference  
on High Performance Networking (HPN'98)  
Vienna, Austria, September 21 - 25, 1998

*edited by*

**Harmen R. van As**  
*Institute of Communication Networks*  
*Vienna University of Technology*  
*Vienna, Austria*



SPRINGER SCIENCE+BUSINESS MEDIA, LLC

---

**Library of Congress Cataloging-in-Publication Data**

A C.I.P. Catalogue record for this book is available  
from the Library of Congress.

---

ISBN 978-1-4757-5397-4      ISBN 978-0-387-35388-3 (eBook)  
DOI 10.1007/978-0-387-35388-3

**Copyright © 1998** by Springer Science+Business Media New York  
Originally published by Kluwer Academic Publishers in 1998

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher, Springer Science+Business Media, LLC

*Printed on acid-free paper.*

# Contents

---

Preface	ix
Committees	xi
Reviewers	xiii
<b>Part One: Broadband Internet Access</b>	
Broadband access to the Internet - an overview <i>H. Leopold (Invited speaker)</i>	3
Performance of multiple access protocols in geo-stationary satellite systems <i>H. Koraitim, S. Tohmé, M. Berrada, A. Brajal</i>	25
A new HFC architecture using return path multiplexing <i>J. C. Yee</i>	45
<b>Part Two: Multimedia Multicast</b>	
End-to-end reliable multicast transport protocol adaptation for floor control and other conference control functions requirements <i>N. Kausar, J. Crowcroft</i>	65
An architecture for conference-support using secured multicast <i>T. Hardjono, N. Doraswamy, B. Cain</i>	79
SELDOM: A simple and efficient low-cost, delay-bounded, online multicasting <i>T. Alrabiah, T. F. Znati</i>	95
<b>Part Three: Scalable Multicast</b>	
A scalable and robust feedback mechanism for adaptive multimedia multicast systems <i>A. Youssef, H. Abdel-Wahab, K. Maly</i>	117
A scalable protocol for reporting periodically using multicast IP <i>L. Blazević, E. Gauthier</i>	143

A scalability scheme for the real-time control protocol <i>R. El-Marakby, D. Hutchison</i>	153
---	-----

#### **Part Four: ATM Infrastructure**

Enhanced convolution approach for connection admission control in ATM networks <i>J. L. Marzo, J. Domingo-Pascual, R. Fabregat, J. Solé-Pareta</i>	171
Fast rerouting in ATM networks: Pro-active search protocol <i>I. Lievens, T. Catrysse, P. Demeester</i>	189
Impact of VC merging on buffer requirements in ATM networks <i>A. L. Schmid, I. Iliadis, P. Droz</i>	203
A comparison of ATM stream merging techniques <i>M. Baldi, D. Bergamasco, S. Gai, D. Malagrino</i>	219
Integrating parallel computing applications in an ATM scenario <i>J. Vila-Sallent, J. Solé-Pareta</i>	235

#### **Part Five: Next generation internet**

Differentiated services: A new approach for quality of service in the internet <i>F. Baumgartner, T. Braun, P. Habegger (Invited speaker)</i>	255
Toward a hierarchical mobile Ipv6 <i>C. Castelluccia</i>	275
Active libraries: A flexible strategy for active networks <i>D. C. Lee, S. F. Midkiff</i>	291

#### **Part Six: QoS in the Internet**

End-to-End QoS provisioning through resource adaptation <i>D. G. Waddington, D. Hutchison</i>	309
A dynamic sender-initiated reservation protocol for the internet <i>P. P. White, J. Crowcroft</i>	327
USD: Scalable bandwidth allocation for the internet <i>Z. Wang</i>	351

A connectionless approach to providing QoS in IP networks <i>B. Nandy, N. Seddigh, A. S. J. Chapman, J. Hadi Salim</i>	363
---	-----

## **Part Seven: IP/ATM Internetworks**

An implementation of a gateway for hierarchically encoded video across ATM and IP networks <i>J.-M. Robinet, Y. Au, A. Banerjea</i>	383
--	-----

Trading off network utilisation and delays by performing shaping on VC ATM connections carrying LAN Traffic <i>P. Castelli, L. Guida, M. Molina</i>	399
--	-----

Packet-based approach to ATM cell policing, and their effects on internet traffic <i>C. Song, R. Wilder, T. Dwight</i>	417
---	-----

Optimising bandwidth reservation in IP/ATM internetworks using the guaranteed delay service <i>C. A. Malcher Bastos, M. A. Stanton</i>	429
---	-----

## **Part Eight: Internet Applications**

Orchestra!: An internet service for distributed musical sessions and collaborative music development and engineering <i>P. Bussotti, F. Pirri</i>	455
--	-----

High-performance online presentation of complex 3D scenes <i>S. Olbrich, H. Pralle</i>	471
---	-----

On the optimal placement of web proxies in the internet: The linear topology <i>B. Li, X. Deng, M. J. Golin, K. Sohraby</i>	485
--	-----

The network computer for an open services market <i>L. Henckel, J. Kuthan</i>	497
--	-----

## **Part Nine: Internet Networking**

Integrated Services: IP Networking Applications <i>G. Howard (Invited speaker)</i>	511
---	-----

The interaction of the TCP flow control procedure in end nodes on the proposed flow control mechanism for use in IEEE 802.3 switches <i>J. Wechta, A. Eberlein, F. Halsall</i>	515
---	-----

On end-to-end congestion avoidance for TCP/IP <i>J. Martin, A. Nilsson</i>	535
---	-----

## **Part Ten: Flow and Congestion Control**

A rate based back-pressure flow control for the internet <i>C. M. Pazos, M. Gerla</i>	555
--	-----

TCP-BFA: Buffer fill avoidance <i>A. A. Awadallah, C. Rai</i>	575
--	-----

Motivation of an end-to-end regulation of bandwidth in intra-networks: The ROBIN concept <i>M. Frank, P. Martini</i>	595
---	-----

Nondeterministic classifier performance evaluation for flow based IP switching <i>J. Karvo, M. Ilvesmäki</i>	613
---	-----

## **Part Eleven: QoS Routing and Scheduling**

Internet QoS routing using the Bellman-Ford algorithm <i>D. Cavendish, M. Gerla</i>	627
--	-----

Feedback controlled scheduling for QoS in communication systems <i>J. Schiller</i>	647
---	-----

Scheduling algorithms for advance resource reservation <i>C. Xu, J. W. Wong</i>	659
--	-----

Achieving 90 % throughput in a flow-oriented input-queued switching router system <i>G. S. Kuo, P.-C. Ko</i>	673
---	-----

Service logic mobility over intelligent broadband networks <i>Ch. Z. Patrikakis, S. E. Polykalas, S. S. Venieris</i>	685
---	-----

# Preface

Under the theme of *'The Millenium Push of Internet'* the eighth IFIP Conference on High Performance Networking (HPN '98) is taking place at the Vienna University of Technology, Vienna, Austria, September 21 - 25, 1998. Its successful earlier conferences were held in Aachen, Germany (1987), Liège, Belgium (1988), Berlin, Germany (1991), Liège, Belgium (1992), Grenoble, France (1994), Palma de Mallorca, Balearic Islands, Spain (1995) and White Plains, New York, USA (1997). The conference series has been established to be a forum where specialists from industry, network operating, and academia can share their experiences in the newest trends in high performance networking. Thereby can high performance be viewed as high-speed, high throughput, low delay or a quality measure representing characteristics such as high flexibility, high availability, high scalability, high functionality or high security.

HPN '98 focuses on the manifold hot issues related to the revolutionary evolution of Internet and Intranets. Today all forms of multimedia networking start playing an important role. Designed as computer networks, the traditional interconnecting networks were used to carry computer data only. Now, Internet and Intranets are also used for Internet telephony, multimedia conferencing, remote lecturing, distributed simulations, network games and many real-time applications more. Future application areas, with even more sophisticated real-time requirements, might include telemedicine, distributed workgroups, distance learning and telecommuting. Multimedia networking faces many technical challenges like real-time data over non-real-time networks, high data rates over limited network capacities, and unpredictable availability of network capacity. Therefore, protocols for real-time applications, routing, congestion and flow control, quality-of-service management, interworking with ATM, high-performance end systems and applications, as well as multicast protocols are focal topics of this conference.

The program of HPN '98 consists of three invited papers highlighting main trends in the Internet arena and 38 papers selected from 69 paper submissions based on three reviews per paper. The papers are presented in eleven sessions during three days. Two days with half-day tutorials proceed the conference. Special thanks are due to IFIP WG 6.4, to the members of the organizing and technical committees, to the Vienna University of Technology, to the sponsors of the conference, to the staff of the Institute of Communication Networks, and particularly to Mrs. Johanna Pfeifer for her very successful and highly appreciated organizational support.

Harmen R. van As      Conference Chair

Vienna, September 1998



## Committees

---

### INTERNATIONAL PROGRAM COMMITTEE:

General Chair:

Harmen R. van As, Vienna Univ. of Technology, A

Ian Akyildiz, Georgia Tech, USA

Torsten Braun, Univ. of Berne, CH

Augusto Casaca, INESC, Portugal

Andre Danthine, Univ. of Liege, Belgium

Michel Diaz, Univ. Toulouse, France

Christophe Diot, INRIA, France

Otto Duarte, Univ. Fed. Rio de Janeiro, Brazil

Jörg Eberspächer, Techn. Univ. Munich, Germany

Serge Fdida, Univ. Paris VI, France

Zygmunt Haas, Cornell University, USA

Marjory Johnson, NASA-RIACS, USA

Paul Kühn, Univ. Stuttgart, Germany

Ralf Lehnert, Dresden Univ. of Technology, Germany

Helmut Leopold, Alcatel, Austria

Kurt Maly, Old Dominion Univ., USA

Olli Martikainen, Helsinki Univ. of Technology, F

Georg Mittenecker, Vienna Univ. of Technology, A

Hussein Mouftah, Queens Univ., Canada

Ignas Niemegeers, Univ. of Twente, The Netherlands

Guru Parulkar, Washington Univ. St. Louis, USA

Stephen Pink, SICS, Sweden

Radu Popescu-Zeletin, GMD Fokus, Germany

Ramon Puigjaner, Univ. Illes Balears, Spain

Guy Pujolle, Univ. Versailles, France

Doug Shepherd, Univ. Lancaster, UK

Thomas Sommer, Siemens Austria, A

Otto Spaniol, Univ. Aachen, Germany

Ralf Steinmetz, Techn. Univ. Darmstadt, Germany

Ahmed Tantawi, IBM Res., Yorktown Heights, USA

Fouad Tobagi, Stanford Univ., USA

Samir Tohmé, ENST, France

Giorgio Ventre, Univ. of Napoli, Italy

Martina Zitterbart, Univ. Braunschweig, Germany

LOCAL ORGANIZING COMMITTEE:

Harmen R. van As (Chair)

Reinhard Kuch, Post & Telekom Austria

Helmut Leopold, Alcatel Austria

Georg Mittenecker, Vienna Univ. of Technology

Johanna Pfeifer, Vienna Univ. of Technology

Karl Prucha, Ericsson Austria

Reda Reda, Siemens Austria

Thomas Sommer, Siemens Austria

## Reviewers

---

Joseph Bannister  
Matthias Baumann  
Kemal Bengi  
Torsten Braun  
Igor Brusic  
Georg Carle  
Augusto Casaca  
Sungrae Cho  
Christian Cseh  
Christophe Diot  
Stefan Dresler  
Otto Duarte  
Jörg Eberspächer  
Serge Fdida  
Reinhard Fleck  
Matthias Frank  
Maurice Gagnaire  
Marnix Garvels  
Carsten Griwodz  
Robert Haas  
Hossam S. Hassanein  
Vesna Hassler  
Ilias Iliadis  
Joe Inwheel  
Marjory Johnson  
Val Jones  
Matthias Kaiserswerth  
Ahmed E. Kamal  
Ulrich Killat  
Ralf Lehnert  
Wolfram Lemppenau  
Helmut Leopold  
Michael Liepert  
Antonino Lucantonio  
Kurt Maly  
Olli Martikainen  
Peter Martini  
Lorne Mason

Thomas Meuser  
Enzo Mingozzi  
Georg Mittenecker  
Yan Moret  
Torsten Müller  
Victor F. Nicola  
Stefan Noll  
M. Serafim Nunes  
Philippe Owezarski  
Rudolf Pailer  
Sergio Palazzo  
Erich Plasser  
Reinhard Posch  
Aiko Pras  
Christian Prehofer  
Antoni Przygienda  
Ramon Puigjaner  
Quernheim  
Josef Rammer  
Günter Remsak  
Eveline Riedling  
Kassem Saleh  
Henning Sanneck  
Jon Schuringa  
Doug Shepherd  
Achim Steinacker  
Ralf Steinmetz  
Burkhard Stiller  
Ahmed N. Tantawy  
Samir Tohmé  
Dirk Trossen  
Harmen R. van As  
Teresa Vazao  
Alexander Wachlowski  
Adam Wolisz  
Johnny W. Wong  
Ellen Zegura  
Martina Zitterbart

# **Part One**

---

## **Broadband Internet Access**

# Broadband access to the Internet – an overview

*H. Leopold*

*Alcatel*

*Scheydgasse 41, A-1211 Vienna, Austria,*

*Tel.: +431 27722 3551, Fax.: +431 27722 1172,*

*E-mail: [helmut.leopold@aut.alcatel.at](mailto:helmut.leopold@aut.alcatel.at)*

## Abstract

The evolution of the modern telecommunication environment depends on three main factors: (i) the market demand for new services, (ii) the politically stimulated liberalisation of the telecommunication market, and (iii) the technological advances of new telecommunication technologies. The technological advances include getting the fibre closer to the subscribers, cheaper equipment costs and standardisation.

Network operators are forced nowadays to enable, by offering new services, very high profit in a very short period of time with limited investment. For this reason, especially the access network and the technologies linked to it have an increasing importance for the economic success of an operator in a liberalised telecommunication market.

This article describes the most distinguishing aspects of an access network, its positioning within a telecommunication system, and the most important technological developments in this field: Fibre In The Loop (FITL) systems, Digital Subscriber Line (xDSL) technologies on copper twisted pair and the Hybrid Fibre Coax (HFC) technology on CATV networks.

## Keywords

**Access network, fibre in the loop (FITL), xDSL, ADSL, HFC, CATV, cable telephony, cable data modem**

## 1 INTRODUCTION

The evolution of the modern telecommunication environment depends on three main factors:

- the technological advances of new telecommunication technologies,
- the market demand for new services, and
- the politically stimulated liberalisation of the telecom market.

Especially the liberalisation in the telecommunications market has a tremendous impact on business in this area. New actors like international network operators, Cable TV (CATV) network operators, energy suppliers, railway organisations, city communication operators, etc. will start to offer telecommunication services in competition to the incumbent national network operator. This will have an impact on the market share, the tariff structure, the Quality of Service (QoS) and the offered services to the customers itself. The final target is the so called "Full Service Network (FSN)", which is capable of offering all types of interactive multimedia services at any time, any place and any QoS.

The provisioning of new telecommunication services in general and new multimedia services in particular is made possible by the availability of several new technologies as well as advances in standardisation:

- Optical signal transmission and SDH technology which allows the establishment of powerful backbone networks.
- New switching and transmission technologies like ATM which allows the transport of information, independent of their nature, and which allows dynamic establishment of high-bandwidth connections with flexible Quality of Service (QoS), in order to meet the application requirements. However, the recently initiated discussion concerning the integration of Internet- and ATM-technologies will have an enormous impact on future developments in this field (Sales, 1998).
- Low cost digital mass storage permits economical deployment of digital video servers.
- Powerful video compression techniques like MPEG effectively reduce storage expense and transmission bandwidth demand.
- New wireless technologies for access and transmission networks based on satellite or terrestrial.
- New access network technologies to bring multimedia capacity over the last mile to a broad range of subscribers.

These technological advances are to a large extent also stimulated by the regulatory changes around the world for more open competition, liberalisation, and privatisation. Operators world wide are faced with this changing environment and are now in process to upgrade their network infrastructures in order to become a competitive network operator in the new telecommunication environment.

The question is now, how all these providers will transform their network systems and their organisations. Network operators are forced to maximise the revenue in a very short period of time with a limited investment. For this reason,

especially the access network technologies linked to it have an increasing importance for the economic success of an operator in the liberalised telecom market. The access network represents an essential part of the overall costs of a telecommunication infrastructure. The investment cost are up to 70% for the access network and the ownership of the access network, i.e. the Operation and Maintenance (OAM) is up to 80% of the total cost. OAM costs include preventive and corrective maintenance, network management and repair of defective equipment.

In the past years there has been an international effort from telecommunication operators and manufacturers to maximise consensus on the systems required in the local access network to deliver a full set of telecommunications services, both narrowband and broadband (FSAN, 1997).

This article discusses the positioning of an access network in an overall telecommunication system and presents the recent advances of access technologies, where beside Fibre In The Loop (FITL) systems especially Digital Subscriber Line (DSL) technologies for twisted-pair telephone lines and Hybrid Fibre Coax (HFC) technology for coaxial CATV networks which allow the utilisation of existing access network infrastructures in order to offer broadband services additionally, are of main importance.

## 2 EVOLUTION OF THE ACCESS NETWORK ARCHITECTURE

The general trend in telecommunication networks is to establish powerful backbone networks which are composed of only a few network nodes. These nodes are interconnected by broadband connections and have an extensive access network, reducing the public switching hierarchy levels (Berkowitz, 1997).

All network operators which are running fibre optic networks, have established such powerful backbone network infrastructures by using SDH, Frame Relay (FR) and ATM technologies in order to enter the telecommunication business. Such infrastructures are owned, beside the incumbent operators, by new actors like energy suppliers, railway organisations, city communication operators etc. Now it is important to recognise that such an infrastructure is the basis for entering the business market segment only, since there is no appropriate access network available in order to address the mass market, i.e. the small and medium sized enterprises (SMEs) and the households to offer new multimedia telecommunication services.

Following the ITU-T terminology, an access network is comprising those network entities which provide the required capabilities for the provision of telecommunication services between the local exchange (Lex) and the subscribers; i.e. the related User Network Interface (UNI)<sup>1</sup>.

---

<sup>1</sup> According to ITU-T recommendation G.902 the access network is allocated between the so-called Service Node Interface (SNI) at the V reference point (at the Lex) and the User-Network Interface (UNI) at the T reference point at the subscriber premises (G.902, 1995). The VB5.1 interface provides access for multiple users of an access network to the service node. The VB5.2 interface has the

Especially the liberalisation of the European telecommunication market January 1<sup>st</sup>, 1998 has resulted in a much stronger competition between network operators as well as manufacturers. Thus, also the access network will be influenced by this new framework. New access network technologies have to be able to offer traditional services very cheap and to realise new multimedia services at the same time with a reasonable price. The most important factors for the design of new technologies for the access network are:

- Minimising the Operation and Maintenance (OAM) costs.
- Improvement of the Quality of Services (QoS) by improving the fault and service management capabilities for example.
- Offering of new services: Incumbent network operators are looking for new revenues and are trying to protect their existing customer base. New operators have the intention to gain market shares by offering new services. Thus, the new access network has to support existing narrowband services like Telephony, ISDN and 64 kbit/s to 2 Mbit/s leased line services, as well as new broadband services like fast Internet access and digital video transmission.

Thus, new access networks have to be flexible in order to support the increasing bandwidth demand as well as any type of service.

### 3 TODAY'S PSTN-ACCESS NETWORK

The physical access infrastructure of the classical Public Switched Telephone Network (PSTN) is based on copper twisted pair lines (a/b-lines), which connect the subscribers to the next local exchange. In most industrial countries around 99 % of the households are connected by copper twisted pairs to the PSTN in order to offer the classical narrowband telecommunication services. Only in rural areas, wireless access systems are installed as well. Today wireless access networks are investigated in general to offer mobility as an additional service value to the subscribers.

From a main distribution trunk within the local exchange (Lex), cables with some 400 to 600 copper wires are used to finally connect the subscribers with single twisted pairs in a point-to-point star architecture. A typical distance from the Lex to the subscriber is around 4 km. In rural areas some longer distances are used of course.

The PSTN was designed to provide telephony services to the subscribers. The early users needed only the set-up of connections to different locations, so they could talk to each other. This is the Plain Old Telephone Service (POTS). Basically, the provision of POTS is still the rationale behind the telephone network and operators still generate most of their incomes from POTS services.

---

additional capability of controlling the access network from the service node via a dedicated protocol in such a way that it is possible to have concentration within the access network on a per call basis.



A digital service, offered today over the PSTN, is realised by the Integrated Services Digital Network (ISDN). ISDN is offering a common interface for voice and data services up to 2 Mbit/s.

Further on, the PSTN today is also used for different data services, like the access to the Internet, by using voice-band modems with capacities of up to 56 kbit/s through the 4 kHz voice channel.

As already mentioned above, network operators still generate most of their income from POTS services. However, the need for new more sophisticated services based on Internet traffic will change the importance of the role of the PSTN. This new situation will have a tremendous impact on the architecture and technology of the new access network architecture.

## 4 USE OF FIBRE IN THE ACCESS NETWORK

Optical fibre in the access network is a prerequisite to enable the necessary bandwidth for a final full service network. However, the key question is: how far in the access network is it economically viable to go.

Dependent on the use of fibre in the access network, and thus the location of the optical network termination, i.e. the location of the so called Optical Network Unit (ONU), we have the following network architectures:

- FTTH (Fibre To The Home): The fibre is terminated at the customer premises.
- FTTB (Fibre To The Building): In this case the ONU is located in the basement of the served building, and may be up to 500 m from the customer apartment or office.
- FTTC (Fibre To The Curb): In this case the ONU may be up to 500 m from the customer premises. A longer distance between the customer and the ONU requires a much less fibre deployment. Such architectures are also called Fibre to the Cabinet (FTTCa), respectively Fibre to the Node (FTTN), Fibre To The service area (FTTSA) and Fibre to the last amplifier (FTTLA) in CATV networks.

The trend is to bring fibre nearer and nearer to the home in order to increase the capacity, improve the reliability as well as get very low error rates of the network infrastructure. However, FTTH have been proved very expensive, due to the high cost of civil works and the low customer share of optics and electronics equipment.

The new installation of cables, independent whether fibre or copper, is – if no appropriate duct is available – the largest cost factor in a network. FTTC/FTTB offers a good compromise, taking advantage of the high bandwidth of fibre, while utilising the existing asset of the local loop in the last kilometre. In FTTB/FTTC network architectures copper twisted pair or coaxial cable is used to cover the last section to the terminal equipment. The different technological solutions are discussed below.

Optical networks are supporting the transmission of analog as well as digital signals. The network topology of an optical network is either based on a point-to-point configuration or a point-to-multipoint configuration which might result in a

tree-structure. Based on the use of electrical amplification of the signal within the network we have to differentiate between an Active Optical Network (AON) and a Passive Optical Network (PON).

An optical network (AON or PON) has to be considered as the most promising approach to achieving large-scale full service access network deployment that could meet the evolving service needs of network users. A PON is able to cover some 20 km. This distance depends on the number of used splitting points. A passive optical splitter is generating some power loss which results in a limited distance. Within an AON the optical signal will be amplified within the network which results in a much longer distance. In addition the AON offers additional concentration functions due to the switching capabilities at the splitting points.

## 5 ACCESS TECHNOLOGIES FOR MULTIMEDIA SERVICES

### 5.1 The technology choices

Depending on the physical media used for the access network, different technologies will be employed (ETSI, 1995, Griffith, 1996):

- Twisted-pair copper lines using Digital Subscriber Line (xDSL) technologies like HDSL, ADSL, and VDSL;
- Coaxial cable by using HFC (Hybrid Fibre Coax) technology, where especially the use of fibre in the access network plays an essential role;
- FTTB/FTTC network architectures using copper twisted pair or coaxial cable to cover the last section to the terminal equipment. A combination of fibre/copper is usually called a FITL (Fibre In The Loop) system. A FITL infrastructures is mainly used to offer narrowband services.  
Will the fibre be deployed up to the subscriber, a Fibre To The Home (FTTH) access is realised.
- B-ISDN fibre access lines<sup>2</sup>. Since in such a scenario the fibre is always going up to the subscriber, we get a FTTB or a FTTH architecture. On such access lines, usually Metropolitan Area Network (MAN), Frame-Relay or ATM networks are connected.
- Wireless systems, terrestrial or by satellite.

### 5.2 Green field situation or use of the existing access infrastructure

By the development of new access technologies, it is important to note that there are practical and sensible compromises between cost, functions and performance. The choice of the appropriate access technology is based on two factors:

- Services to be offered: narrowband or broadband services, services for households and SMEs or services for large business users.

---

<sup>2</sup> B-ISDN UNI specifications are available for 155 Mbit/s, 622 Mbit/s, 1.5 Mbit/s, 2 Mbit/s, 51 Mbit/s, and 25.6 Mbit/s; see ITU-T recommendation I.432.1-5.

- Deployment of a new access infrastructure or reuse of an existing access infrastructure. In a so called „green field situation” there is no existing access infrastructure available. However, such a situation is not very often the case in industrial countries.

In a green field situation FITL- or HFC-networks are deployed usually. HFC technology will be used if video broadcast is part of the offered service. A FITL access infrastructure is appropriate if narrowband services like POTS, ISDN and leased line services up to 2 Mbit/s are of main importance. If a quick service offering of narrowband services and mobility is of main importance, wireless technologies are appropriate.

The owners of access infrastructures (copper or coax) are driven to capture profits from new services. In this way, they might try to invest as little as possible and go for immediate results. In this approach, the linearity of the investment against the possible return is important. This is even more true looking to the uncertainty of market share for new services. Thus the reusability of the already available access infrastructure is of main importance. For the reuse of an existing physical infrastructure (coax cable or copper twisted pair), the following technologies have to be considered:

- HFC-technology, if video broadcast services are part of the offered services.
- FITL-technology if some part of the existing copper infrastructure has to be upgraded by fibre and the offered services are mainly POTS, ISDN and nx64 kbit/s (up to 2 Mbit/s) leased line services.
- xDSL-technologies, if broadband services to the subscriber have to be offered via the point-to-point twisted-pair copper lines.

The different technological solutions are discussed in the following.

### **5.3 Direct fibre access (FFTH/FTTB) and Fibre In The Loop (FITL) systems**

As highlighted above, a direct broadband fibre access (FTTH/FTTB) which offers an enormous capacity for the offering of multimedia services, is economically just feasible for very large business customers, which are requesting symmetrical access and high capacity to a backbone network. However, recent standardisation activities are aiming at developing very cost effective optical network terminations (FSAN, 1997). Such a cheap optical network termination is the prerequisite for realising any cost effective FFTH infrastructure in the future.

A FITL (Fibre In The Loop) system is a combination of fibre cables feeding neighbourhood Optical Network Units (ONUs) and last mile connections by existing or new copper wires. A FITL infrastructure is mainly used to offer narrowband services like POTS, ISDN and leased line services up to 2 Mbit/s. The Deutsche Bundespost Telekom project in the eastern part of Germany is a good example of a large FITL deployment, based on a PON architecture, in a green field situation (Hytas, 1995).

The access technologies which are based on the reuse of existing infrastructure and which are the basis for offering multimedia services on a broad range to the subscribers are discussed in the following sections: xDSL and HFC.

## 6 DIGITAL SUBSCRIBER LINE (DSL) TECHNOLOGIES

### 6.1 xDSL Technologies: HDSL, ADSL, VDSL

It is important to note that ISDN was the first technology to transport digital signals up to 2 Mbit/s over the twisted pair access. The new available Digital Subscriber Line (DSL) transmission techniques like HDSL (High bit rate Digital Subscriber Line), ADSL (Asymmetric Digital Subscriber Line) and VDSL (Very high speed Digital Subscriber Line) can deliver data at multi-Mbit/s over the unscreened, twisted telephone wires, originally intended for bandwidths of between 300 Hz and 3,4 kHz. This is due the remarkable advances in digital signal processing technology. These allow the implementation of sophisticated channel modulation techniques which suggest that there are no fundamental technological barriers to overcome, at least on the digital side. In fact the challenges are mainly of an analog nature, set by the external electrical environments; i.e. the imposed noise interference which will degrade the Signal-to-Noise Ratio (SNR).

For the development of these new technologies, emphasis is not only given to maximum bitrate possibilities, but also on the robustness of the physical transport medium as well as different service characteristics. For that reason, different modulation techniques are used throughout the solutions.

#### *HDSL (High bit rate Digital Subscriber Line)*

HDSL realises the transmission of 2 Mbit/s over a copper twisted pair in both directions (upstream and downstream) and offers usually a standardised G.703 interface to the subscriber. To improve the max. possible distance for the transmission of the 2 Mbit/s, 2 or 3 copper twisted pairs can be used in parallel. Since the bandwidth capacity is the same in both directions, this is also called a SDSL (Symmetric Digital Subscriber Line) technology. HDSL is becoming a well accepted technology to offer especially for business customers 2 Mbit/s leased line services<sup>3</sup>. The use of the HDSL technology for the realisation of a multimedia application is described in (Leopold, 1996).

#### *ADSL (Asymmetric Digital Subscriber Line)*

An ADSL system connects an ADSL modem pair through a twisted pair copper line, creating a high bit rate downstream channel and a medium bit rate upstream channel (Chen, 1994). The high bit rate downstream channel ranges from 1,5 – 7,5 Mbit/s, while the upstream channel ranges from 16 to 640 kbit/s. This is achieved without disturbing the POTS service already installed on the line. The POTS

---

<sup>3</sup> Also 64 kbit/s leased line services are technically feasible; however there is no clear market requirement for such a service up to now.

compatibility is achieved by using a higher frequency band for the digital signals than used for the analog telephone signal (i.e. above 4 kHz up to 1 MHz). The analog signals are separated from the digital signals by a so called „POTS splitter“.

The downstream high-speed channel is based on the assumption that most residential high-speed services will be asymmetric. The business users requiring symmetric services will install fibre for bi-directional data transfer. The downstream data rates depend on a number of factors, including the length of the copper line, its wire gauge and cross-coupled interference. Line attenuation increases with line length and frequency and decreases as wire diameter increases.

### *VDSL (Very high speed Digital Subscriber Line)*

VDSL is a complimentary development of the ADSL technology. By use of VDSL a much higher bandwidth will be achieved by a much less distance to the customer: up to 25 Mbit/s in downstream direction and up to 2 Mbit/s upstream direction at distances of up to 1 km. This dimensioning makes the VDSL technology to a good extension of a FTTCa-architecture. However, VDSL is still in the definition phase and further developments have to be expected.

### *ISDL (ISDN Digital Subscriber Line)*

A further xDSL variant is to integrate ISDN and ADSL. However, such an ISDN Digital Subscriber Line (ISDL) approach is only useful for the integration with an ISDN basic access. The ISDN-primary access (2 Mbit/s) is usually used to connect PABXs with the PSTN. The ADSL technology has a complete different objective: to bring multimedia capacity to the subscriber (i.e. up to 6 Mbit/s). However, standards are not finalised yet, and the market acceptance of such an integration has to be verified very carefully.

### *ADSL-Lite*

In order to improve the market acceptance, a special ADSL variant has been developed. ADSL-Lite has the objective to reduce the installation effort at the subscriber premises by allowing a simple plug in by the subscriber in any wall-outlet in the home, just as usual base-band modems. This is achieved by eliminating the POTS splitter, but results also in a compromise concerning the ADSL performance.

## **6.2 Modulation Techniques**

In fact modern coding and modulation techniques provide a level of performance that approaches the theoretical limit of the physical bandwidth. New ADSL developments are using Discrete Multi-Tone (DMT) modulation techniques. DMT is a form of multicarrier modulation. In the frequency domain, DMT divides the channel into a large number of sub-channels. Only those frequency channels will be used which are not disturbed. These channels will be identified during the initialisation phase and will be permanently checked again during operation in order to adapt to a changing environment (i.e. highly robust in noisy environments). This technique is called rate adaptive ADSL (RADSL).

Another technique, the so-called Carrier Amplitude Phase (CAP) modulation technique, is under discussion within standardisation as well. However, DMT is an accepted technology and thus standardised within ANSI T1E1.4.

### 6.3 ADSL and ATM Integration

Modern ADSL products, are integrating ATM and ADSL technology. The main advantages by such an architecture in the access network are the following:

- Support of different traffic characteristics (continuous bit streams, data bursts, etc.);
- Any granularity of bit rate for different user channels.
- Multiple QoS levels per subscriber (ATM connections), allowing a service range from low cost residential to premium business at a guaranteed quality level.
- Combining DMT and ATM provides a flexible bit rate, exploiting the maximum line transfer capacity.
- The offering of different interface types. Beside Ethernet for the traditional connectivity of PCs and Network Computers (NCs), the ATM-Forum 25,6 Mbit/s has been established as a new broadband standard interface at the users premises.

### 6.4 ADSL System Architecture

The ADSL access network encompasses the ADSL modems and the access multiplexer system at the local exchange (Lex) and the ADSL modems at customer premise connected via the local loop. The ADSL modem at the Lex side is also called the ADSL-LT (line termination) and the modem at the subscriber premises is called ADSL-NT (network termination).

The access multiplexer system and the modems at the Lex side are usually combined into a single unit called the access node (using the ADSL Forum terminology) and also referred to as the “DSLAM” (DSL Access Multiplexer) or Subscriber Access Multiplexer (SAM). When the backbone network is ATM, the access node is connected to an ATM access switch. The ADSL access node and ATM access switch may or may not be co-located. The function of the ATM access switch is to concentrate and switch traffic from a number of access nodes onto the regional broadband network. The ADSL access node (DSLAM) performs the following functions:

- Line Termination (LT) of the ADSL subscriber lines.
- Concentration/multiplexing of the ADSL subscriber lines towards the broadband network. WAN interfaces such as a STM-1 are expensive resources for an operator. It is important to concentrate as many subscriber lines as possible onto a single network interface. A multiplexing scheme that provides high concentration while guaranteeing the individually negotiated QoS will be an important asset for network operators, because it will allow them to offer differentiated services at a reasonable cost.

- Termination of customer ATM signalling channels. To provide a standard, scalable mechanism for supporting switched virtual circuit service to ADSL customers, the ADSL access node should terminate the ATM signalling protocol from each ADSL customer, and generate a single standard UNI signalling interface toward the access ATM switch.

At the subscriber premises the installation of a network termination (ADSL-NT) is required to which a LAN, a PC, a Network Computer (NC) or a TV set with an appropriate Set Top Box (STB) is connected.

As shown in Figure 3, the subscriber has now the usual PSTN access for POTS services, an access to the Internet and On-line services to an Internet Service Provider (ISP) as well as an access to digital video contents.

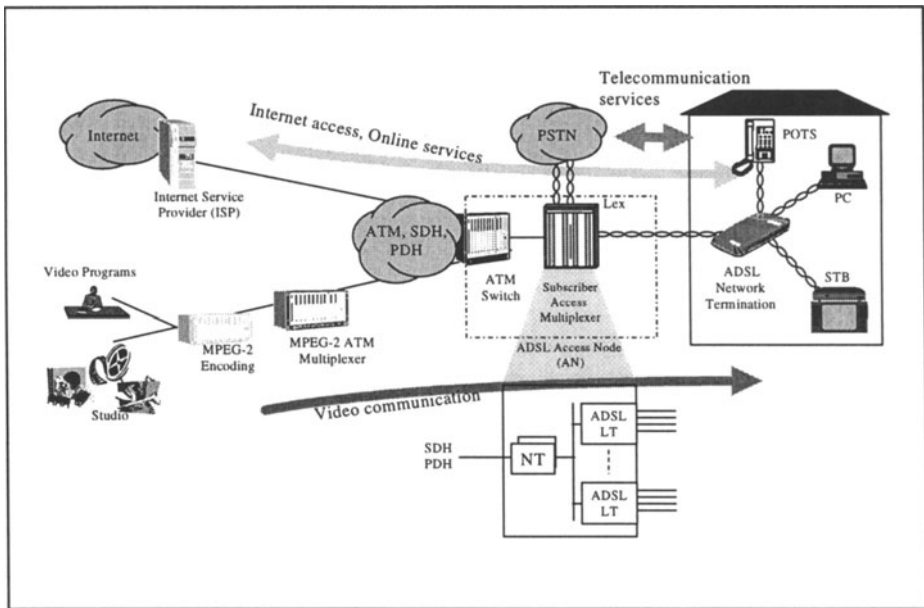


Figure 1. ADSL network architecture

## 6.5 The business opportunities of xDSL Technologies

To summarise, the business opportunities of xDSL-Technologies are:

- Use of the existing PSTN access network resulting in low installation cost which allows the quick connection of new users and allows early service deployment.
- Costs are strictly proportional with the number of connected customers. Thus, a low upfront investment is necessary at the one hand, which limits the financial risk of the operator.

- Sufficient capacity for multimedia services with no impact on the telephony service at the same line.
- Bandwidth asymmetry is well suited for typical client/server applications.

## 7 CATV-OPERATORS AND HYBRID FIBRE COAX (HFC) ACCESS NETWORKS

### 7.1 The CATV business

Basically the broadcast TV business is „entertainment“. Thus, the customer is not really willing to pay for the transmission of signals, but he is prepared to pay for the content he receives. Comparing to the traditional telecom business, other revenues than transmission tariffs are necessary. Thus the prime revenue streams are based on advertising, licenses, basic cable fees, and pay TV. However, it has to be noted, that the traditional TV business is a low margin business with long return of investments.

It seems to be obvious that CATV operators have to protect first of all their core business against Direct Broadcast Satellite (DBS) services especially stimulated by the new digital TV services, while exploring new business areas like data communications, Internet access, online services, and to look for new market shares in the classical telecommunication business to ensure growing revenues in the long term.

However, the main problem that CATV operators have to consider is the move in mindset needed to take advantage of the new market opportunities. Entering the new telecommunication business means more than just a technological upgrading of the network infrastructure. It also means the definition of new services and further on establishing relationships with customers through marketing, customer service, etc. CATV operators have to learn to think more like telecommunications companies. Such a need for cultural change is a potential critical factor for CATV operators.

### 7.2 The new business opportunity of a HFC network

Traditional CATV networks make extensive use of coaxial copper cable to carry the signal from the head-end (HE) to the subscriber. Along the path amplifiers are deployed to compensate for the attenuation of the long coax cable lines. Each broadcast TV signal takes 6 MHz for a NTSC signal or 8 MHz for a PAL signal here in Europe. The bandwidth available on these pure coax networks usually range from 300 to 500 MHz.

During the 80s and 90s cable operators began upgrading their networks to a Hybrid Fibre Coax (HFC) architecture to provide higher quality, increased programming, and new services. These new networks combine high capacity fibre lines with inexpensive coax lines and are the basis for establishing bi-directional capabilities.



HFC based networks have Broadband Optical Network Terminations (BONTs) that typically serve some hundred to some thousand homes by a coax cable; i.e. serving area of a so called “coax cell”. 500 homes per coax cell are seen as the optimised configuration to offer the whole set of possible services from analogue broadcast TV to professional telecommunication services. Such a modern HFC network is able to offer the following set of services for example:

- 40 analogue TV broadcast channels,
- 200 broadcast digital TV channels,
- 400 on-demand channels (Near Video On Demand (NVOD, etc.)),
- Internet access via the TV set,
- Internet access and high-speed digital data services via LANs/PCs, and voice telephony services.

Moving to full service providers, digital TV, data communications, Internet access, on-line services, and telephony services, allow CATV operators to grow their business by

- protecting their core business against the DBS competition,
- entering new business areas,
- competing with the traditional telecom operators, and
- offering the access for the „last mile“ for other network operators which don't have direct access to the customers.

### 7.3 Services to be offered on CATV networks

There are four service classes to be offered on CATV networks. Figure 2 shows the different functional blocks at the subscriber premises<sup>4</sup>:

#### *Analogue TV broadcast*

This is the traditional means of transmitting analogue TV signals to subscribers. Analogue audio distribution and information services via Teletext are established as a well accepted services by the subscribers. The Teletext service is based on the distribution of information (text and graphics) in a broadcasting communication mode without any user interactivity over the network.

Based on these broadcast services, either analogue or digital, first interactivity is realised basically by using the PSTN POTS service. Thus new services like Home Order Television (HOT) which allows the so called “impulse tele-shopping” applications, and information on demand services based on Teletext are offered.

Pay-TV services based on a simple scrambling of the analogue TV signals, is one of the new services being offered by many CATV operators. For transportation of the control information from the subscriber to the CATV operator, usually the POTS service of the PSTN is used. For such a Pay-TV service, the customer requires an analogue STB in order to decode the scrambled TV signal<sup>5</sup>.

---

<sup>4</sup> The functional modules do not necessarily relate to implementations; i.e. several functional modules might be combined within one implementation in the future.

<sup>5</sup> E.g. The Pay-TV channel Premiere.

### *Digital TV and interactive multimedia services*

For digital TV broadcast based on the Digital Video Broadcast (DVB) standards and for interactive multimedia services based on the Digital Audio Video Council (DAVIC) specifications, the subscriber requires a DVB compliant digital STB in order to decode the digital MPEG-2 encoded video streams. Digital TV broadcast (audio and video) is basically offering the same services as for the analogue signal transmission. However, due to the digital transmission a very much higher capacity in the network is achieved. Dependent on the coding schema and the used bandwidth for the digital video signal<sup>6</sup>, this allows the transmission of approximately 8 digital channels per 8 MHz<sup>7</sup>. Thus Near Video on Demand (NVOD), and distribution of selected programs to a specific set of customers are the major services planned. For such services, no interactivity over the network is required.

If the CATV network is offering bi-directional functionality, the communication from the customer to the CATV operator is done via the CATV network, otherwise the use of the PSTN is possible as well.

Based on such interactive services, Pay per View (PPV), Internet access, on-line services and interactive multimedia services can be offered to the subscriber (Furth, 1995).

### *Cable telephony services*

Real interactivity based on a return channel within the CATV infrastructure is necessary for the offering of telecommunication services like POTS, ISDN and 64 kbit/s to 2 Mbit/s leased line services. These services can be seen as „classical telecom services“, offering guaranteed QoS, which are implemented over the CATV infrastructure by a so called „cable telephony“ system (Fletcher, 1997). All services are based on 64 kbit/s channels. At the subscriber premises a so called Coax Network Termination (CNT) is installed to offer the different interfaces for POTS, ISDN and the nx64 kbit/s leased line services. The bandwidth capacity is usually below 2 Mbit/s in both directions. This service class allows the CATV operator to get into the lucrative telephony business.

### *Data communication services*

These service class is based on offering LAN interfaces at the subscriber premises like Ethernet in order to allow Internet access and online-services by the use of a PC and LAN interconnection to realise intranetworks. This is implemented by a so called „cable data modem“ system.

It is important to note, that the users of a cable data modem systems are using a shared medium (i.e. a legacy LAN), which means, that the offered bandwidth of the 10 Mbit/s Ethernet for example, have to be shared by all users related to this LAN (Laubach, 1996). Like in a legacy LAN environment, there are no QoS guarantees provided by the network.

---

<sup>6</sup> The transmission bandwidth required (e.g. 2, 4, 6 or 8 Mbit/s), depends on the intended video quality.

<sup>7</sup> By the use of a QAM-64 modulation, 32 Mbit/s can be transported within a 8 MHz channel.

During the last few years much effort has been made by international organisations and companies to build suitable physical and access protocols (MAC) for interactive services using the CATV network for data communications and the development is still ongoing. Within the European Digital Video Broadcasting (DVB) Project the Return-Channel Working Group has developed a complete specification for a physical and MAC protocol in cooperation with the Digital Audio Council (DAVIC).

In North America several large cable companies have joined forces to build the Multimedia Cable Network Systems (MCNS). MCNS have set up the Data Over Cable Service Interface Specification (DOCSIS). IEEE has created a working group 802.14 that is also developing a standard for a physical and a MAC protocol. The MCNS is focusing on the pure Internet environment and is advanced very well, while the DVB/DAVIC and the IEEE activities are focusing on the use of ATM. Which standard will succeed on the market finally is still not clear.

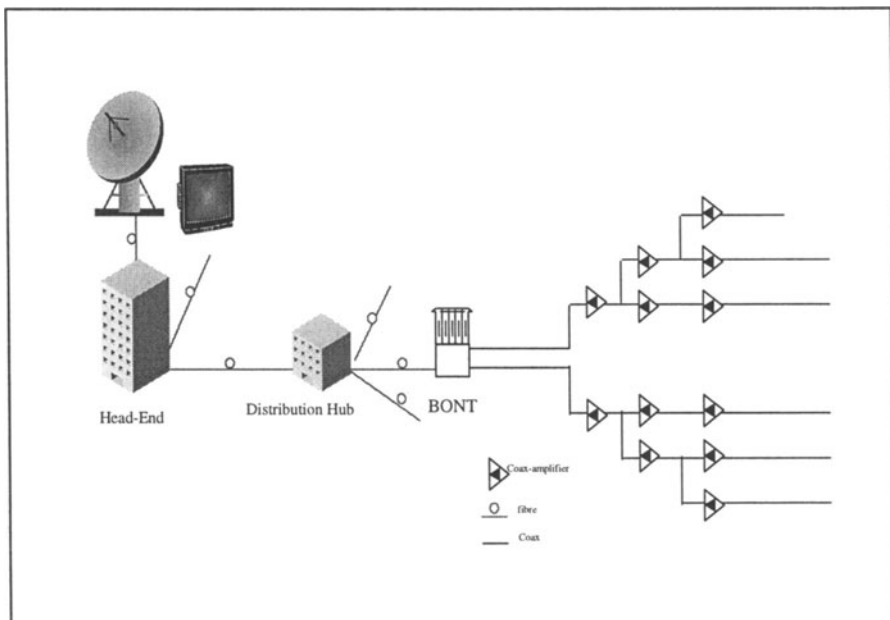


Figure 2: Services over the CATV infrastructure

### *ATM on HFC Networks*

In order to support the different traffic characteristics by a flexible communication architecture, the use of ATM in HFC networks is investigated within the EU ACTS project ATHOC. For more details refer to Böttle, 1997a and Böttle, 1997b.

## 7.4 Evolution of CATV to modern Telecommunication Networks: HFC

To allow the realisation of new services, CATV network infrastructures have to be upgraded in order to provide new capabilities for the subscribers. The starting point for the CATV operators are the existing coax architecture. Purely coaxial CATV networks evolve to Hybrid Fibre Coax (HFC) networks, by implementing

- a bi-directional network, with capacity for interactive services,
- new electrical amplifiers for the coax cable infrastructure, working up to 800 MHz to offer enough capacity for all services, and
- a fibre overlay network to improve the reliability and to bring the transmission capacity as near as possible to the customer.

### *HFC - Physical Network Infrastructure*

The Hybrid Fibre Coaxial (HFC) architecture has become a standard for CATV operators. This architecture does not contain switching elements in the distribution network, and only requires optical-to-electrical conversion, amplification and power splitting. Thus every customer receives the same signal, which contains the information of all the services provided.

A typical HFC network has a fibre star point-to-multipoint subnetwork and a tree-and-branch coaxial subnetwork. The Head-End (HE) receives, modulates and transmits the CATV channels over the fibre network towards the Broadband Optical Network Termination (BONT), where the signals are converted back to electrical signals to serve the customers by coaxial cable. Thus, from a physical point of view, a HFC network is made of two parts:

- An optical network, which runs from the optical transmitter at the network head-end to a point close to the customer, in a star or multiple star architecture. The fibre part of the HFC network is a Passive Optical Network (PON) or an Active optical network (AON) usually with two fibres, one for each direction of transmission (also called „two fibre system“). Using optical fibres to a large extend, instead of the coax infrastructure is improving in addition reliability and maintenance expenses.
- A coaxial branch-and-tree network, which connect the termination of the optical network (BONT) with each customer. The BONT performs the transition between multiple main coax cable and the optical fibre. If the serving area of the coax infrastructure is up to 250 m only, no amplifiers are used. This is the recommended approach for green field situations.

Typically some intermediate points between the head-end and the BONTs have been introduced which allow a hierarchical structuring of the fibre distribution network; so called Distribution Hubs (DH) (see Figure 3). These nodes perform optical signal amplification and optical signal splitting. These nodes could be seen as equivalent to the local exchanges of the telephone networks and might be the points where new equipment for the transmission of new services will be placed

(further obvious locations for the new equipment are the HE as well as the BONT<sup>8</sup>).

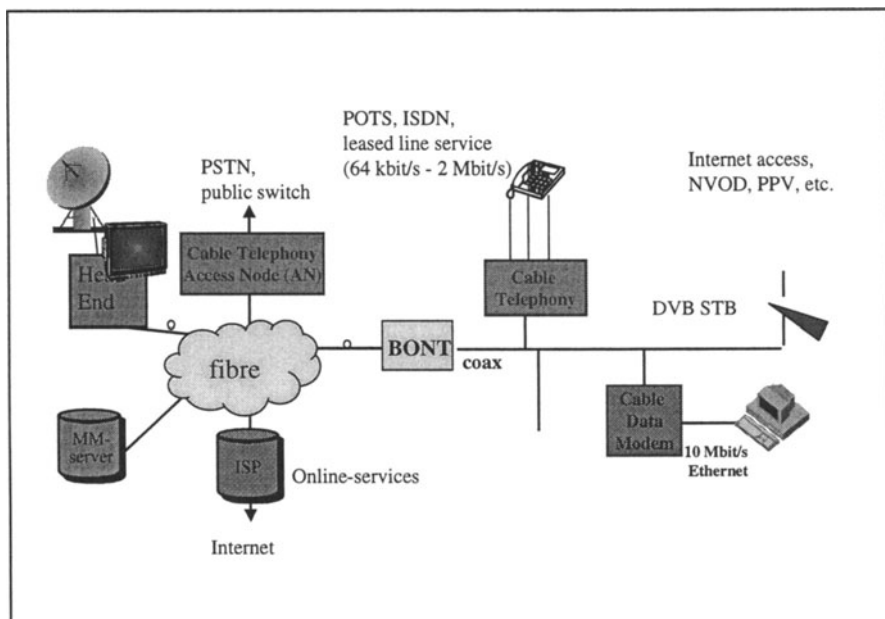


Figure 3. HFC network architecture: HE, DH, BONTs, coax amplifiers (LE)

### Signal transmission

A HFC network transports the information (analogue and digital signals) by using frequency multiplexed analogue carriers (FDM). Each TV channel is modulated on a different Radio Frequency (RF) carrier; allocating 8 MHz per analogue TV channel (European standard; PAL).

In a traditional CATV network, usually the transmission is just done uni-directional from the network to the user (downstream) by a broadcast communication mode (since the very first usage of this infrastructure is analogue video distribution). This limitation is usually due to the uni-directional amplifiers within the coax network. These amplifiers are working in the frequency range up to 302, 446, 600 or 860 MHz. After upgrading or replacing all the uni-directional coax-amplifiers, the infrastructure is able to transport the signals bi-directional. However, the signals transmission is not symmetrical.

The upper part of the spectrum is used for **downstream** transmission and the signal is distributed to all subscribers in a broadcast communication mode. Encryption and/or other measures to control the access to the information have to be used for services other than broadcast TV to ensure appropriate security. The

<sup>8</sup> Although usually there is no space for additional equipment available at the BONT location.

lower part of the spectrum is used for **upstream** transmission. A dedicated protocol is required for sharing the available capacity (see below).

### *Upstream signal transmission (return channel)*

The return channel employed will have significant impact on the services offered and their degree of interactivity. Although it is not imperative to implement the return channel in the same network as the downstream channels, doing so leads to a homogeneous and user friendly solution.

The upstream signals that must be transmitted are: network supervision signals, alarms, etc. coming from coax amplifiers, BONTs, DHs, or other network elements dependent on the service implemented on the HFC network, in order to integrate the HFC part in a powerful network management system, and signals for the interactive services like voice channels for telephony services, upstream channels for data services, return channel for interactive multimedia services.

In the CATV network the physical medium - the coaxial cable - is shared by all subscribers. Therefore, an access procedure is required which provides collision free access to the shared medium and assigns capacity according to the customer's requirements. Time division multiple access (TDMA) and frequency division multiplex (FDM) techniques are used to allow the return path to be shared among the subscribers.

The return channel of the CATV network has some impairments which has to be considered for the final system design:

- Limited bandwidth (some 50 MHz only) and modulation with low efficiency due to the usage of a robust modulation technique (a typical spectral efficiency of 1-2 bits/Herz).
- Noise accumulation in the upstream direction from each customer termination with a coax cell. Common sources for external interference are radio signals coupled to the network, engine based interference, TV receivers, etc.

### *Downstream signal transmission*

The downstream channel has a distribution architecture. No noise accumulation from different terminations is performed. For analogue signal transmission, the same modulation technique for terrestrial TV and FM radio services is used for compatibility reasons. For digital services (telephony, data communications, digital TV), a modulation technique providing high spectrum efficiency can be used. Typical examples are 64-, 32-, 64-, 128-, 256-QAM. These modulation techniques provide a spectrum efficiency up to 6 bits/Herz, thus resulting in several Mbps on a 8 MHz channel<sup>9</sup>.

According to the current standardisation discussions, 41 Mbit/s can be transmitted by means of 64-QAM in an 8 MHz wide CATV channel. Considering the redundancy needed for error protection (by a 204,188) Reed-Solomon code, 38 Mbit/s remain for the transmission of user data.

---

<sup>9</sup> Capacity in a 8 MHz channel: 16-QAM: 25 Mbps; 32-QAM: 32 Mbps; 64-QAM: 38 Mbps; 128-QAM: 45 Mbps; although it has to be noted that these figures varies from the different product implementations.

## 8 CONCLUSION

Based on the studies conducted by the FSAN (FSAN, 1997), it was determined that the per-line cost of producing a full service access network will slowly decrease with volume of production. However, at a sufficiently high volume level the development of new technologies becomes justified that can enable significant reductions in per-line equipment and installation costs.

However, no one knows for certain what path is socially optimal for residential adoption of broadband technology. Whatever the predictions, business decisions must be made in both the public and private sectors to determine the next step. Telcos and CATV operators have to continue to upgrade their networks both to achieve network efficiency and to compete in the near term for new and advanced communication services (Potts, 1997). Especially the new operators have to consider:

- The installation of a wireless access system suffers from the limited capacity not supporting multimedia applications currently.
- A new installation of an access infrastructure might be too expensive and thus economically not feasible.
- The use of the copper lines of the PSTN access infrastructure from the incumbent operator (“unbundling”) depends on the interconnection fees.
- The incumbent operators have a powerful potential access infrastructures by the use of xDSL technologies.
- The use of the available CATV infrastructure might be an economical solution.

The access network has a very essential role within a telecommunication infrastructure, and thus it brings an added value which is much more than just the transmission of information between the subscriber and the backbone network. Next steps should aim at demonstrating the availability of technologies and products for broadband access networks realising full service capabilities. This should also contribute to the effort in standardisation to enhance the definition for further system developments.

Finally, it is important to note that only at a sufficiently high volume level the development of new technologies becomes justified. Thus a powerful access network infrastructure, which supports multimedia group communication services to a broad range of users; i.e. households, SMEs and business units, will be the prerequisite for any mass deployment of future multimedia services.

## 9 REFERENCES

- Berkowitz, P. (1997) The Changing Shape of the Access Network. *TELECOMMUNICATIONS*, Vol. 31, No. 10, October 1997 (pp 109-114).
- Böttle D. (1997a) Network Architecture for an ATM based Hybrid Fibre Coax System and related Applications. *CWAS'97 International Workshop on Copper Wire Access Systems*, Budapest, Hungary, October 27-29, 1997.
- Böttle, D. and Wahl, S. and Sierens, Chr. And Bastos J. and Borges I. and Frei P. and Christ, P. and Fahner H. and Ramlot G. (1997b) ATM Applications over Hybrid Fibre Coax Trials. *ISS'97*, Toronto, September 21-26, 1997.
- Chen, W.Y. and Waring, D.L. (1994) Applicability of ADSL to Support Video Dial Tone in the Copper Loop. *IEEE Communications Magazine*, May 1994 (pp.102-109).
- ETSI (1995) Video on Demand Network Aspects. *ETSI/NA5, DTR/NA-52109*, Technical Report, October 1995.
- Fletcher, M. (1997) Cable Telephony: Coming to Market. *TELECOMMUNICATIONS*, Vol. 31, No. 9, September 1997 (pp 81-83).
- FSAN (1997) Full Service Access Network (FSAN) Gx Initiative, summary of results, 25 February, 1997.
- Furth, B. and Kalra, D. and Kitson F.L. and Rodriguez, A.A. and Wall, W.E. (1995) Design Issues for Interactive Television Systems. *COMPUTER*, May 1995.
- G.902 (1995) Access Networks – Architectures, Services, Arrangement and Service Node Aspects. *ITU-T Recommendation, COM 13 –R41*, July 1995.
- Griffith, M. and Guirao, F. and Van Noorden, L. (1996) Network Evolution for residential Broadband Interactive Services - From RACE to ACTS. *European Conference on Multimedia Applications, Services and Techniques (ECMAST '96)*, Proceedings Part I, May 1996.
- HYTAS (1995) *HYTAS: Ein zukunftsorientierter Netzzugang für Multimediadienste*“, Vortrag zur 36. Post- und Fernmeldetechnischen Fachtagung des VDPI – Multimedia – anbieten, transportieren, anwenden“, ke Kommunikations-Elektronik GmbH & Co, Februar 1995.
- Laubach, M. (1996) To foster residential area broadband internet technology: IP datagrams keep going, and going, and going .... *Computer Communications* 19, 1996 (pp. 867-875).
- Leopold, H. and Hirn, R. The Bookshop Project: An Interactive Multimedia Application Case Study. In *Proc. of the International COST237 Workshop on Multimedia Transport and Teleservices*, D. Hutchison, A. Danthine, H. Leopold, G. Coulson (eds), Barcelona, Spain, November 1996 (Springer Verlag LNCS 1185, ISBN 3-540-62096-6).
- Potts, M (1997) Guideline NIG-G1: Broadband Deployment. *ACTS, Network Integration Chain Group*, September 1997 (<http://ginaiihe.ac.be/>).
- Sales, B. and Dumortier, P. and Van Mieghem, P. (1998) Dual-Mode Routing: A long term Strategy for IP over ATM. *6<sup>th</sup> IEE International Conference on Telecommunications*, Edinburgh, 29.3-1.4.1998.



## 10 BIOGRAPHY

Helmut Leopold, born April 27th, 1963, in Hohenems, Austria, made his degree in 1982 in Electronic and Communications at the Technical College in Rankweil, Austria. In 1989 he made the degree of Dipl.-Ing. Informatik (Computer Science) at the University of Technology of Vienna. From 1989 to 1994 he was responsible for the group „Multimedia Communications“ at Alcatel Austria Research Center in Vienna and was actively involved in international standardisation and in European R&D projects in the broadband communication area. Since 1994, Mr. Leopold is with Alcatel Austria AG, extending his activities on marketing and usage of new multimedia services based on broadband technologies. Since 1996 he is account manager for the CATV-market in Austria.

# Performance of multiple access protocols in geo-stationary satellite systems

*Heba Koraitim, Samir Tohmé †*

*Marouane Berrada, Americo Brajal ‡*

*† Ecole Nationale Supérieure des Télécommunications*

*46, rue Barrault 75634, Paris-France,*

*Tel.:33 1 45817449, Fax.:33 1 45891664*

*e-mail: koraitim, tohme@res.enst.fr*

*‡ Laboratoires d'Electronique Philips S.A.S.*

*22, avenue Descartes - BP 15 - 94453 Limeil Brevannes Cedex France,*

*Tel.:33 1 45106700, Fax.:33 1 45106960*

*e-mail: berrada, brajal@lep.research.philips.com*

## **Abstract**

Two packet multiple access schemes, the DQRAP and the ARRA are modeled and evaluated in the geostationary satellite environment. A new protocol is afterwards proposed joining the advantages of both studied schemes, and more adapted to interactive multimedia applications over satellite uplinks. The Generalized Retransmission Announcement Protocol, GRAP, re-groups the immediate access by contention at low loads, and the reservation access for higher loads to achieve a better channel efficiency. Simulation results illustrate an improved throughput/delay characteristics and a higher protocol stability. Enhanced versions of the protocol are also proposed and evaluated to further improve its efficiency, with reasonable additional complexity.

## **Keywords**

VSAT, satellite networks, multiple access protocols, packet switching

## 1 INTRODUCTION

Satellites systems have been recently rediscovered, to complement terrestrial networks in providing a worldwide access to the evolving multimedia services.

Multiple access protocols constitute one of the most important aspects that largely determine the performance of communications networks. They define the means by which a subscriber can establish the contact to the network and hence gain access to its resources. Network subscribers are then competing in a pre-established manner to share the resources of a specific link. Many families of protocols have been proposed in the literature in different network contexts (Lee *et al.* 1983), (Lee *et al.* 1984), (Raychaudhuri *et al.* 1987), (Wong *et al.* 1991) and (Mohammed *et al.* 1994). The features of these protocols differ according to the type of the considered network and its topology.

Satellite networks have specific characteristics which largely influence the features of the multiple access scheme to be adopted. From one side, the inherent broadcast feature of satellite links enables the broadcast of information to all covered users at almost the same time. However, the large round-trip propagation delay characterizing the geostationary satellite environment imposes some limitations in protocol design.

Two multiple access schemes are studied and analyzed in this work, the Distributed Queueing Random Access Protocol (DQRAP), and the Announced Retransmission Random Access (ARRA) protocol. We have decided to compare these two as they rely on the same basic idea of allowing contention access at low loads and switching to reservation access at higher loads. Besides, both protocols separate new arrivals from retransmitted messages to avoid excessive conflict and collisions. In DQRAP (Xu *et al.* 1993, Koperda *et al.* 1995), this is done by blocking new arrivals from entering the system if a collision resolution cycle is in progress. In ARRA, on the other hand, new arrivals are allowed to transmit in a separate part of the frame called the common mini-slot pool (CMP) (Raychaudhuri 1985).

These two protocols differ however in the collision resolution principle applied in each case. While DQRAP adopts a tree-based approach (Capetanakis 1979) for resolving collisions between reservation requests, ARRA protocol randomly arbitrates retransmission requests over the total number of frame slots.

DQRAP was originally proposed for terrestrial hybrid/fiber coaxial networks. We have therefore adapted the access algorithm to the satellite environment before modeling and examining the performance of both protocols on an uplink satellite channel. A new protocol, the Generalized Retransmission Announcement Protocol GRAP, is then proposed which further enhances the channel efficiency and achieves a better system stability at higher loads.

In the next section, the two protocols, DQRAP and ARRA are briefly described, where the DQRAP adaptation to the satellite environment is particularly emphasized. The two protocols are modeled and their simulation results

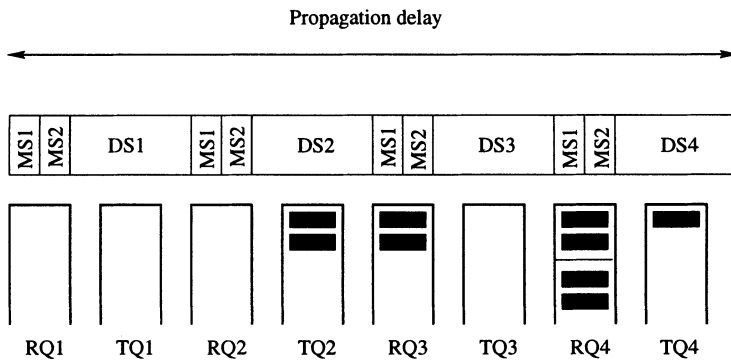
are presented. In section 3, the GRAP protocol is introduced, and detailed, explaining the modifications added to enhance its performance. Section 4 reveals the protocol models developed to extract the performance evaluation results which demonstrate the behavior of GRAP and E-GRAP and compare it to other simulated protocols.

## 2 PROTOCOLS IN THE SATELLITE CONTEXT

Due to the large round-trip propagation delay encountered in satellite links, the channel is always structured in the form of frames, whose length can vary to cover a part of, or the entire propagation time. In other words, the frame duration may be one or a multiple of channel frame durations. For both DQRAP and ARRA, the frame is divided into a number of equal length slots, where each slot is further divided into a data transmission slot (DS) and a number of control mini-slots (MS).

### 2.1 Satellite DQRAP

The frame structure of the DQRAP protocol is shown in figure 1. The frame duration is equal to the two-way round-trip propagation time from a user terminal to the network control center (NCC) and back again to the terminal. The status of the system is broadcasted to all terminals within the satellite



**Figure 1** DQRAP frame structure for satellite links

coverage by the NCC. Each active terminal keeps track of the channel status by listening to the feedback information of the NCC. A distributed queueing discipline is maintained by each terminal to memorize the status of each slot, and keep track of its own position in the queues. This discipline functions by storing the status information of each slot in two global queues, the transmission queue,  $TQ$ , and the collision resolution queue,  $RQ$ .

When a terminal generates a packet, it will listen to the information concerning the queues status of the upcoming slot after the packet arrival. If both queues are empty (slot 1 in figure 1), the packet will be transmitted in DS together with a reservation request transmitted in a randomly selected MS of the same slot.

If the DS is successful, the corresponding reservation will be canceled. Otherwise, if the DS collides, while the MS is successful, the terminal then enters the (TQ) of the corresponding slot on the frame, and keeps track of its position in the distributed queue to transmit its data in the same slot of the next frame after the round-trip propagation delay.

However, if both, the DS and MS suffer collisions (more frequent at higher loads), the terminal will be placed in the *RQ*, in the same position with all other terminals colliding in the same MS of the corresponding DS. Colliding requests are organized in the *RQ* by the order of the position of their MS in the slot (Collisions in MS 1 enter in first position, while those in MS 2 enter in the second position).

The first group in the *RQ* will retransmit only their request in the same slot, after randomly selecting another MS. Request collisions in a certain slot are hence resolved according to a tree algorithm by progressively dividing colliding groups in mini-slots into smaller groups. Collision resolution trials are then separated by the round-trip propagation delay (or the frame duration), as they are always attempted in the same slot of the frame.

Whenever a packet arrives in a terminal and learns, by the feedback information, that the *RQ* of the upcoming slot is empty, while the *TQ* is not (slot 2 in the figure), the terminal transmits only a contending reservation request in one of the MSs of this slot. If however, the *TQ* is empty, while the *RQ* is not (slot 3), or if both are non-empty (slot 4), the packet will be held in an arrivals queue in the terminal until a slot on the frame is detected with an empty *RQ*.

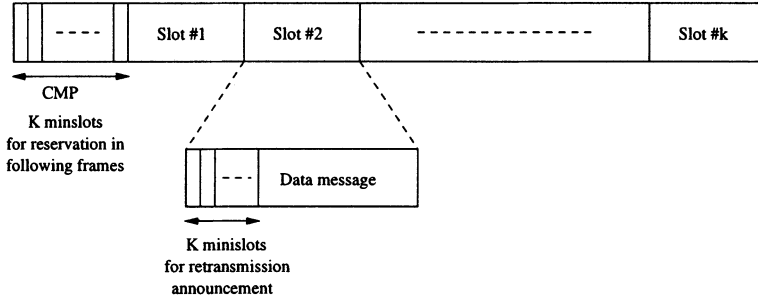
An occupied *RQ* indicates that a collision resolution cycle is in progress in this slot, and hence new arrivals are prohibited from entering the system. This measure serves to decouple new arrivals transmissions from retransmissions, thus increasing the protocol stability.

## 2.2 Satellite ARRA

In contrast to the DQRAP protocol, which was originally proposed for terrestrial networks, the ARRA protocol initially targeted broadcast channels with a large round-trip propagation delay. The frame structure of ARRA is shown in figure 2, where the round-trip delay consists of a multiple of frame durations.

The frame is divided into a number  $K$  of slots and each slot in turn, is divided into a data slot (DS) and  $K$  mini-slots (MS). Similar to the DQRAP

protocol, the length of an MS is much smaller than the length of a DS. Both, the DS and the MS, have the same roles as in DQRAP for transmitting the data packet and the reservation requests respectively. An additional field of  $K$  MSs constitute the first part of the frame to serve as a Common Mini-slot Pool (CMP).



**Figure 2** ARRA frame structure for satellite links

When a packet is generated, the terminal will wait for the beginning of the next frame to read the feedback information transmitted by the NCC concerning the frame. If there are free slots available for contention on this frame, it will choose a free slot at random and transmit its packet in the DS field of the slot. A reservation request is also simultaneously transmitted in one of the MSs of the same slot, to advertise the need for reservation if the data message collides. The MS within the slot is randomly chosen and the selected MS number indicates the slot in which retransmission will take place in the following frame, after the round-trip propagation delay. Each terminal is then *announcing* a reservation in case of collision.

If no free slots are available on the frame, a terminal will only transmit a reservation request in one of the MSs of the Common Mini-slot Pool (CMP) at the beginning of the frame, to reserve the corresponding slot number in the upcoming frame. The feedback information regarding the status of the frame and the available slots for contention is broadcasted by the NCC before the beginning of each frame.

No special measures are adopted in this protocol to resolve collisions between retransmissions, except for the fact that each back-logged terminal in a collision will randomly choose a different slot each time by announcing in the corresponding MS number. This randomizes the retransmission trials over the whole available set of free slots on the frame.

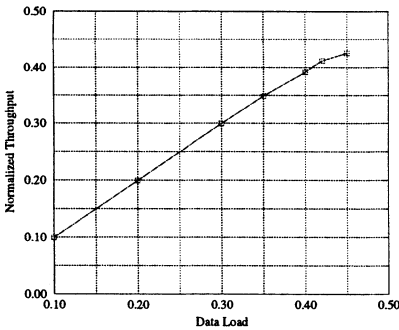
### 2.3 Performance of DQRAP and ARRA

The two protocols were modeled in a VSAT star network configuration, where a satellite with on-board processing capabilities ensures the NCC access control functions. A Poisson model was considered to model the traffic generated by an infinite number of VSAT user terminals.

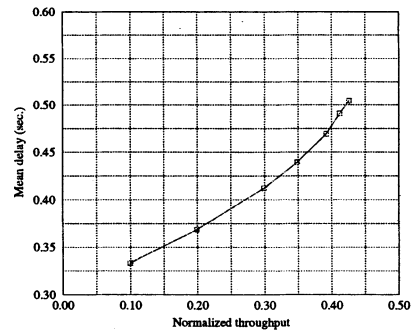
DQRAP protocol was simulated with a frame duration equal to the round-trip propagation delay (0.27 sec.), consisting of 108 slots, which is also the number of interleaved DQRAP engines. On the other hand, a frame duration of 20 ms was considered for ARRA protocol, with 8 slots per frame ( $K = 8$ ). This frame duration and number of slots per frame were chosen as a compromise, between the frame length and the overhead introduced by the mini-slots. Both, DQRAP and ARRA, have the same slot length, and were tested for the same uplink channel capacity.

DQRAP protocol was modeled for two values of MS, the first equals three, while the second equals eight. The latter value was particularly considered in order to compare its performance to that of ARRA, when they both have the same number of MSs per slot. The additional overhead introduced by the MSs is not counted for in the results, as we are only considering the throughput of the data slots.

We have noticed, as illustrated in figure 3, that the throughput of the ARRA saturates at a value of 0.425. This is due to the excessive number of collisions taking place between retransmissions and newly arriving reservation requests, in the limited number of mini-slots of the CMP. As the network loading exceeds 0.43, an unstable state is reached where collisions are multiplying and the system output saturates. The throughput/delay characteristics of the ARRA protocol are shown in figure 4.



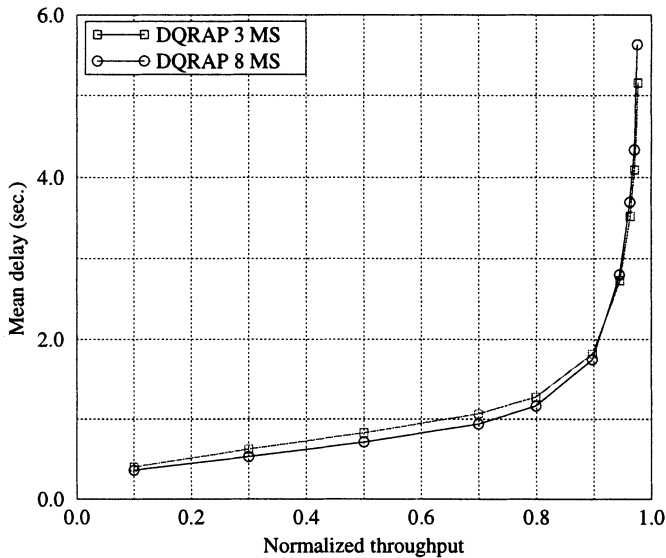
**Figure 3** ARRA throughput



**Figure 4** ARRA characteristics

The performance of DQRAP with the two values of MS (3, 8) is presented in figure 5. Obviously, the DQRAP protocol largely outperforms the ARRA

protocol in terms of maximum channel throughput, which reaches 0.98. This is due to the complete separation of the transmission and collision resolution processes characterizing DQRAP.



**Figure 5** DQRAP throughput/delay characteristics

The protocol is also stable, over a wide range of loading conditions, due to the application of the tree-based collision resolution scheme. It is noticed that increasing the number of MSs per slot from 3 to 8 has a slight effect on reducing the average data delay at lower channel loads, but has no significance at higher loads, as the two curves coincide at a load of 0.9.

### 3 GENERALIZED RETRANSMISSION ANNOUNCEMENT PROTOCOL

The Generalized Retransmission Announcement Protocol, GRAP, combines some aspects of the previously described protocols to develop an access scheme adapted to the satellite characteristics, and at the same time achieving an acceptable efficiency and stability.

The frame structure of GRAP is shown in figure 6. Similar to the ARRA, the frame is divided into  $K$  slots, and each slot consists of a data slot field (DS) and  $K$  mini-slots (MS) field. The frame duration is chosen such that the



overhead introduced by the  $K$  MSs length is very small compared to the DS length.

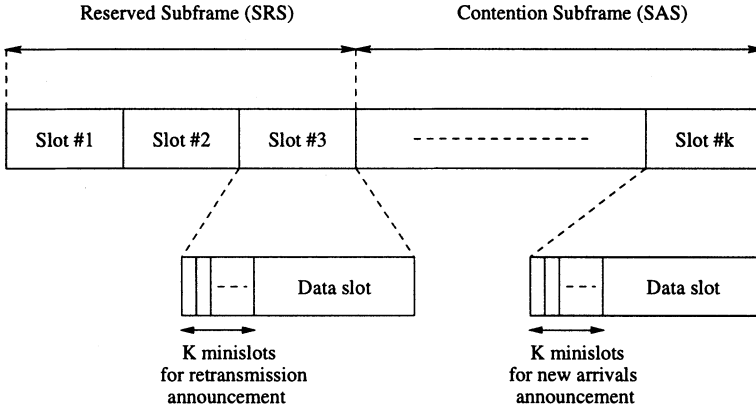


Figure 6 GRAP frame structure

### 3.1 GRAP Access Procedure

*New arrivals* When a user terminal becomes active, it will monitor the down-link stream, to read the status of the upcoming frame and the set of available slots (SAS) broadcasted by the NCC. At the beginning of the following frame, if there are free slots available for contention, the terminal will transmit its data message, accompanied by an anticipating retransmission announcement placed in a randomly chosen MS of the same slot. The MS number in the slot indicates the slot number that the terminal requires to reserve in the next frame in case of collision.

*Retransmissions* If an active terminal receives a feedback indicating that its packet has collided, it will search the SAS for the slot number of its announced reservation. If the slot number is not included in the SAS, it assumes that its reservation has been successful and waits for the assigned slot to transmit its data message.

On the contrary, if the announced slot is included in the SAS, it means that the reservation has not been successful due to collisions or other errors. The terminal then retransmits a reservation request in a randomly chosen MS of an already reserved slot, since reserved slots are not accompanied by anticipating announcements. In this case, the MSs associated with the set of reserved slots (SRS) play the role of the CMP field in the ARRA protocol and replace it, hence reducing the overhead associated with the protocol. Furthermore, the

number of MSs now available for request retransmissions is a multiple of the CMP, and this multiple is equal to the number of slots included in the SRS.

This has the advantage that at high loads, when many slots are reserved, there is more space to retransmit reservation requests, and at low loads, more free slots will be available for contention.

### 3.2 Flow control

In order to limit the risk of repeated collisions at high network loads, a flow control mechanism relying on a back-off delay is introduced in the terminals' access procedure. Before retransmitting a collided request, the terminal will have to wait for a certain amount of delay, before attempting the retransmission. This delay is calculated as a function of the number of repeated collisions to which the terminal has been exposed and is directly proportional to it.

Another control aspect is envisaged whenever the network load goes down, and free slots are again available on the frame, where colliding requests are allowed to recycle in the system as new arrivals. The recycling feature is offered to requests that have exceeded a certain number of repeated collisions, to limit the number of conflicts in MSs and hence improve the protocol stability.

The number of repeated collisions, after which retransmissions are recycled as new arrivals in the system, is a parameter of the protocol that has to be adjusted for optimal operating efficiency. The recycling aspect not only reduces the retransmission load in the MSs, but also reduces the retransmission delay and helps in re-partitioning the load between new arrivals and retransmissions.

### 3.3 NCC operation

The NCC contains all the control and monitoring functions to organize and regulate the GRAP procedure. The uplink frame is completely received before any analysis and processing can be started. The NCC then treats the received frame and analyzes it before sending the feedback to all active terminals on the down-link.

1. **Frame analysis:** The NCC first searches the DS fields of the frame slots to detect successful and collided messages. It then proceeds with the detection of collisions in the MSs. Successful reservations associated with successful contention messages are canceled, while those associated with collided messages are memorized, together with successful retransmission requests in the MSs of the reserved slots in the SRS.

The NCC then allocates slots to successful reservations, and eliminates some of those who had a *virtual collision* (those who place their reservation

request in the same MS number of different slots). The elimination process can be done based on a priority or simply a random principle.

The advantage of having a number of MSs equal to the number of slots on the frame, is the reduction of the amount of information carried by each MS. Hence, the NCC is not obliged to specify the terminal ID associated with each reservation. A terminal can just place a signal in the MS and wait to know whether the corresponding slot is included or not in the SAS.

2. **Feedback formulation:** The set of available slots (SAS) is afterwards formulated, including all non reserved slots on the frame. On the down-link, this SAS is broadcasted, with all other feedback information concerning the reserved slots and their position on the frame. Each terminal then receives the result of its transmission trial, after the propagation and the NCC processing delays (which can consist of one or a multiple of the up-link frame duration).

### 3.4 Enhanced GRAP

In order to further enhance the performance of the protocol, a queueing system is added in the NCC to store virtually colliding reservation requests. The expression *virtual collision* designates the conflict occurring between terminals placing their reservation requests in the same MS number of two or more different frame slots. Although these requests can be correctly received at the NCC, their intention to reserve the same slot on the next frame causes the NCC to hold the reservation for only one terminal and cancel the others.

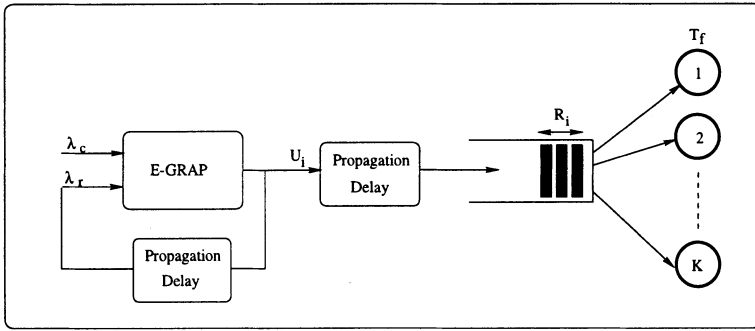
The queueing system then tends to overcome this weakness by preserves the reservations for all *virtually colliding* requests. Two queueing disciplines can be envisaged. The first one comprises a single distributed queue for the whole frame, while the second comprises a number  $K$  of queues, one for each slot on the frame.

#### (a) Single reservation queue

In this scenario, only one reservation queue is maintained by the NCC to store virtually collided requests, for which no place can be found on the upcoming frame, and postpone their allocations for future frames. This can be modeled as shown in figure 7.

The single queue follows a priority discipline, with the higher priority given to requests that have suffered more collisions and hence, more delay. At the NCC, these requests are scheduled at the beginning of the frame and are completely served before new arriving requests. If there remains some place afterwards on the frame, new requests can be accommodated.

Since the remaining free slots on the frame may not correspond to the originally reserved ones for some waiting requests, the NCC must include the



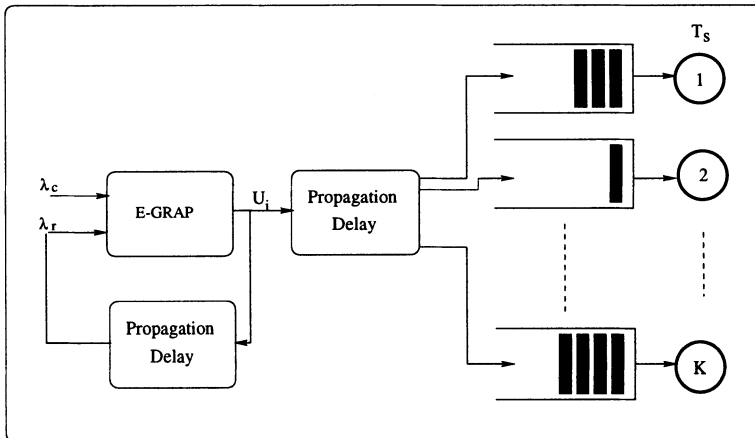
**Figure 7** Single queue E-GRAP model

reserved slot number in its feedback information for each terminal, to notify it of its new reserved slot.

The advantage of this approach is the overall reduction of the mean message delay at higher loads, since the priority is proportional to the number of repeated collisions. A tradeoff has then to be made between the encountered mean delay and the maximum channel efficiency, since more feedback information has to be sent on the down-link.

### (b) Multiple reservation queues

The second scenario envisaged to avoid virtual collisions is the introduction of a number of waiting queues to store successful reservation requests. This number is equal to the number of slots per frame  $K$ , and hence a waiting queue is envisaged per slot as illustrated in figure 8.



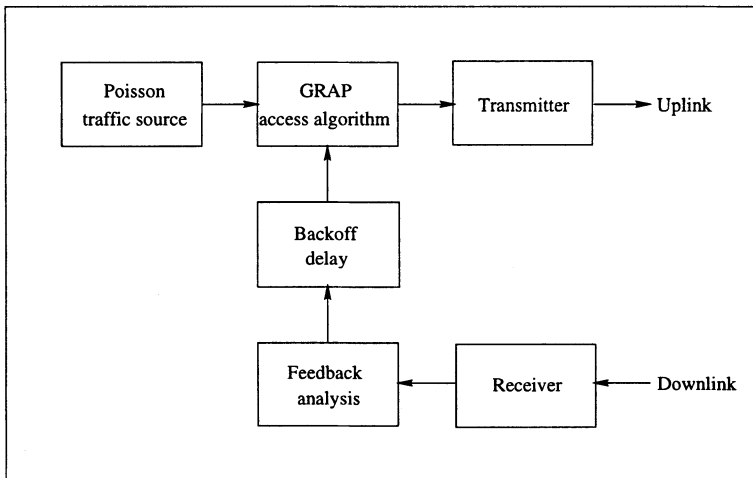
**Figure 8** Multiple queues E-GRAP model

When several terminals transmit their reservation requests in the same mini-slot number of two different slots, all the successful requests are retained and stored in the queue corresponding to the targeted slot. Their position in the queue may either be random, follows their position on the frame, or be decided following a certain priority discipline.

The extra feedback information concerning the reserved slot number is not needed in this approach. This is however, at the expense of the possibility of suffering more delay, waiting for the same slot in future frames, while there may be other free slots on the present frame.

The two approaches explained above help in reducing the number of re-transmissions and hence, possible future collisions. Therefore they confine the delay variation to a smaller value than that experienced by the simple GRAP.

#### 4 GRAP AND E-GRAP PERFORMANCE

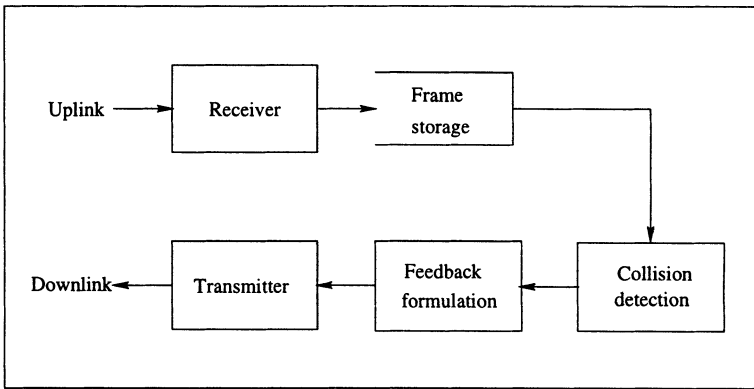


**Figure 9** VSAT terminal model

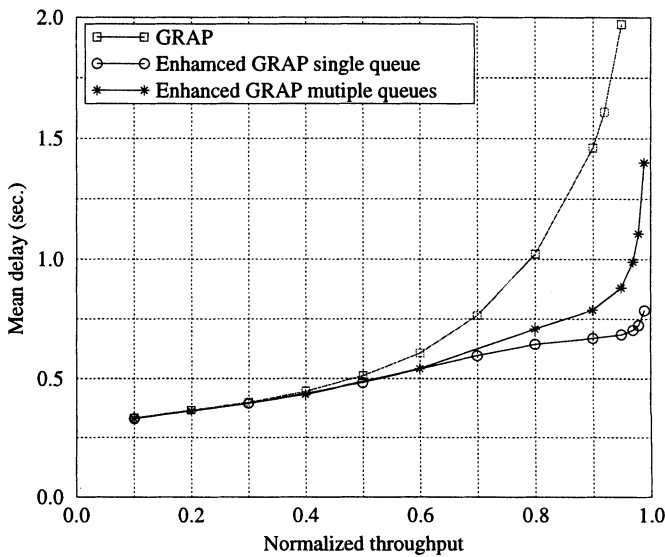
Simulation models have been developed for GRAP and E-GRAP to examine their performance in the same VSAT network configuration, previously described to test DQRAP and ARRA.

Simplified block diagrams for the VSAT terminal model and the NCC model are shown in figures 9 and 10. A frame duration of 20 ms and  $K = 8$  slots per frame have been maintained for the system.

Figure 11 illustrates the throughput/delay characteristics of the GRAP and the two versions of E-GRAP with a single and multiple queues. The GRAP achieves a channel throughput of 0.95 for a mean data delay that approaches 2 seconds. It is clear that, introducing the queueing discipline has improved

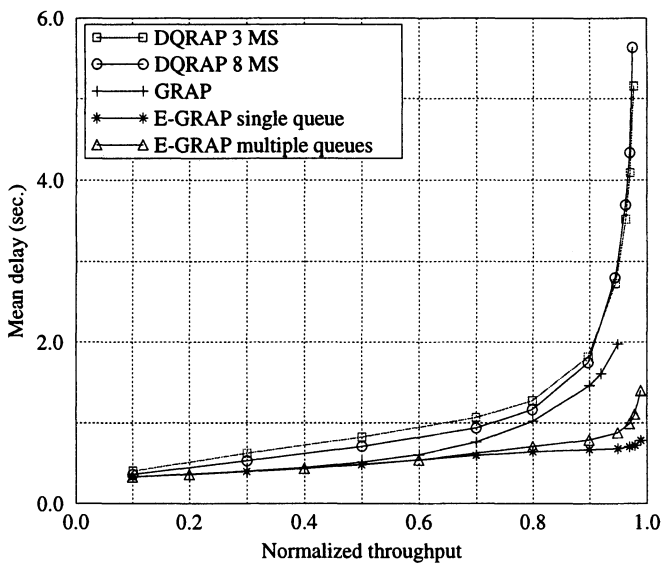


**Figure 10** NCC model

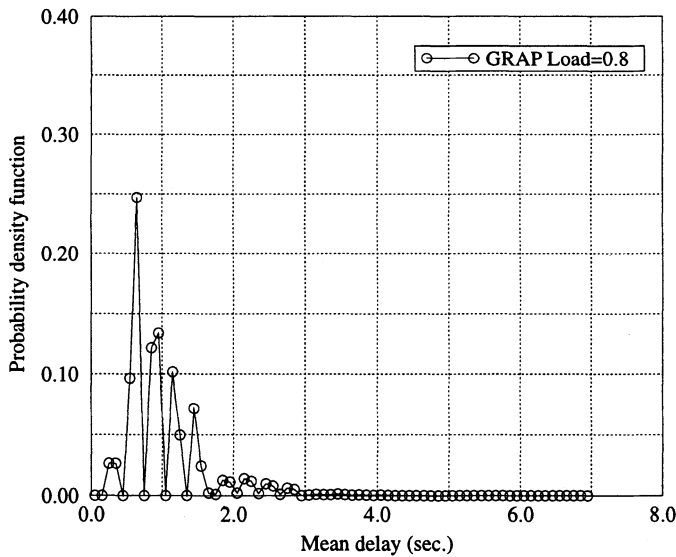


**Figure 11** GRAP and E-GRAP characteristics

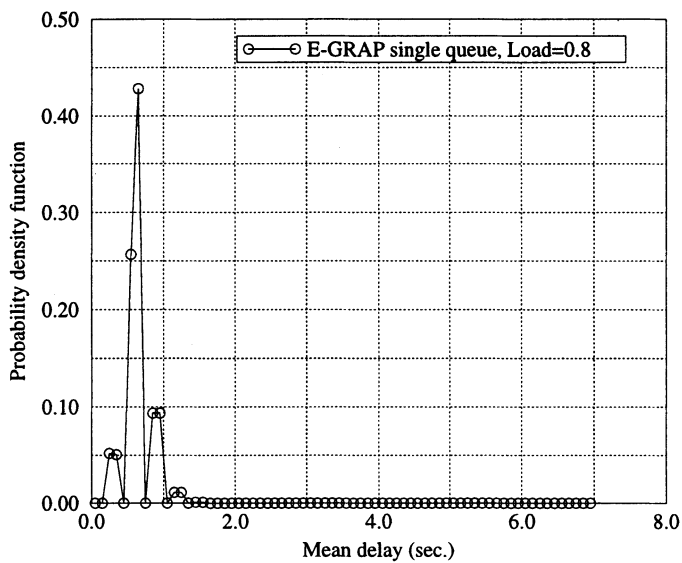
the performance of GRAP by increasing the maximum channel throughput to 0.98, at the same time of reducing the mean data delay in the system. E-GRAP with a single queue further reduces the mean delay at high loads, compared to the multiple queues E-GRAP, due to the complete statistical multiplexing aspect on which it is based. In this version of the protocol then, the number of MS per slot can be different from the number of slots per frame, since a terminal can be reserved any slot on the frame.



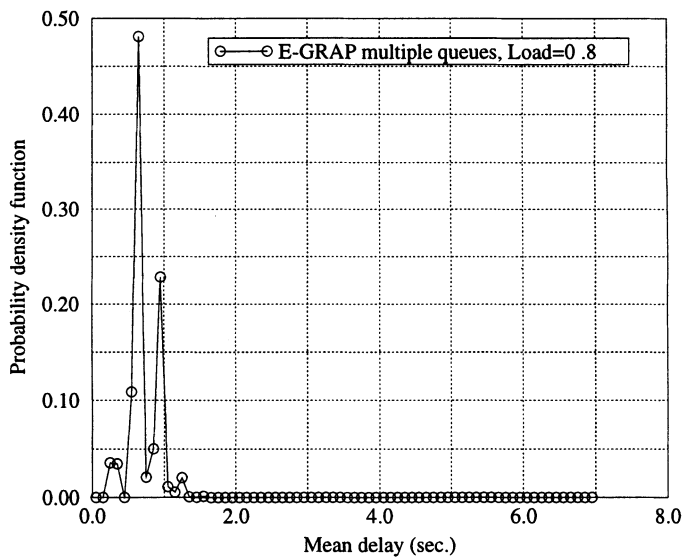
**Figure 12** Performance comparison between DQRAP and E-GRAP



**Figure 13** GRAP delay distribution



**Figure 14** E-GRAP single queue delay distribution



**Figure 15** E-GRAP multiple queues delay distribution



The additional delay introduced by the multiple queues E-GRAP at higher loads, is due to the fact that each terminal must wait for a specific slot that corresponds to the number of MS in which it has previously placed its reservation. The advantage of this technique however, is that the overhead introduced by the MSs in the slot is reduced due to the transmission of less control and feedback information.

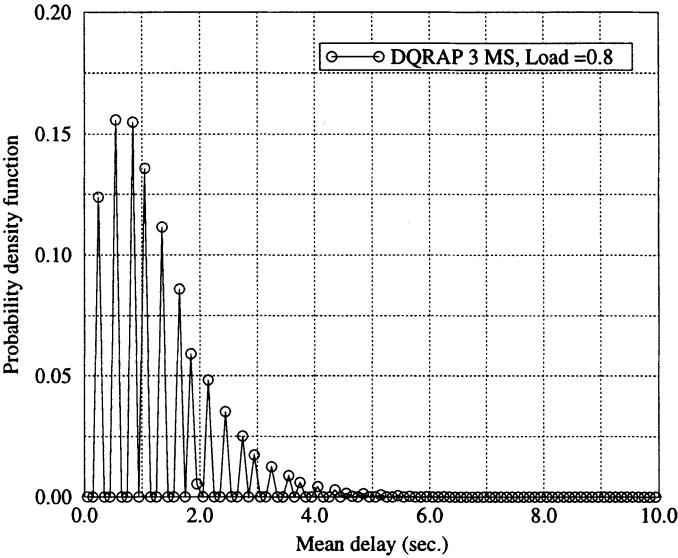
Compared to the DQRAP protocol, as indicated in figure 12, both GRAP and E-GRAP outperform DQRAP, in terms of mean delay, up to a load of 0.95. As the load exceeds the latter value, the improvement introduced by E-GRAP is significant where it attains a throughput value of 0.98 for a delay of only 1.4 seconds. This demonstrates the E-GRAP stability at very high loads.

The delay distribution of GRAP, E-GRAP with single queue and E-GRAP with multiple queues, is measured and illustrated in figures 13, 14, and 15 respectively for a 0.8 loading factor. The E-GRAP is shown to achieve a better delay distribution, where the delay density function is confined to 1.3 seconds at 0.8 load, compared to around 3 seconds for the GRAP.

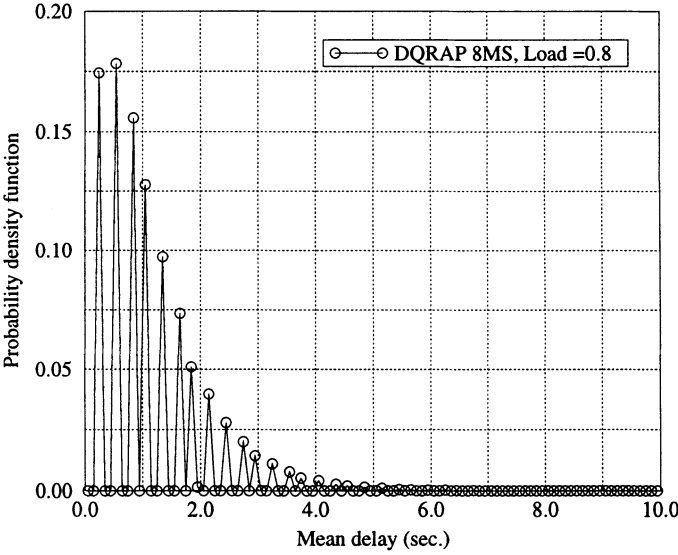
The distribution of the mean delay encountered by data packets in the DQRAP protocol, at the same load (0.8), is given in figures 16 and 17 for 3 and 8 MS respectively. The distributions for both MS values have little difference between them, but they are more dispersed compared to those of GRAP and E-GRAP. The dispersion in the delay values is spread over a wider range reaching a value of 5 seconds. This is due to the tree algorithm applied to resolve collisions of reservation requests in DQRAP.

The results indicate hence, that protocols applying tree-based collision resolution algorithms are stable at high loads, but at the expense of additional delay and delay variation. Besides, the DQRAP protocol introduces more complex functions in the user terminals, since each terminal have to keep track of the state of each of the transmission queue  $TQ$ , and the collision resolution queue  $RQ$  for each slot on the frame.

A back-off delay applied in the GRAP protocol improves the delay performance over that of DQRAP, with much less complexity. The enhanced versions of the protocol, with a queueing discipline, further enhances the protocol performance, at the expense of managing one or multiple distributed transmission queues.



**Figure 16** Delay distribution of DQRAP with 3 mini-slots



**Figure 17** Delay distribution of DQRAP with 8 mini-slots

## 5 CONCLUSIONS

We have modeled and analyzed two packet multiple access protocols in the satellite environment, the DQRAP and the ARRA protocol. The first is based on a tree collision resolution algorithm, which proceeds in parallel to the transmission process of packets in two separate distributed queues for each slot. The second simulated protocol relies on a simple retransmission policy to transmit reservation requests.

A new protocol, the GRAP, was proposed as a compromise between both techniques. It relies on the principle of complete separation between new arrivals and retransmissions. New arrivals access the contention sub-frame, while the retransmissions contend in the mini-slots of the reservation sub-frame. The protocol, however, preserves the right for retransmissions to recycle as new arrivals when a certain number of collisions is encountered.

The frame structure of the ARRA has then been adopted for the proposed GRAP, after removing the CMP field, and the distributed queueing policy of the E-GRAP was inspired by the DQRAP protocol. GRAP and E-GRAP are hence less complex than DQRAP as the whole system can keep track of just one queue (the reservation queue), while they exhibit a more complex feature due to the recycling aspect. This can be easily handled by VSAT terminals.

GRAP was found to outperform both DQRAP and ARRA, where the enhanced versions achieve a better performance and stability at very high loads. E-GRAP exhibits the lowest delay (and hence delay variation), which makes it better adapted to data traffic with real-time requirements, and multimedia applications, such as interactive on-demand services over Internet.

## REFERENCES

- Lee, H. W. and Mark, J. W. (1983) Combined random/reservation access for packet switched transmission over a satellite with on-board processing: Part I - global beam satellite. *IEEE Trans. on Commun.*, COM-31 (N0. 10):1161-1171.
- Lee, H. W. and Mark, J. W. (1983) Combined random/reservation access for packet switched transmission over a satellite with on-board processing: Part II - multi-beam satellite. *IEEE Trans. on Commun.*, COM-32 (N0. 10):1161-1171.
- Raychaudhuri D. and Joseph, K. (1987) Ku-band satellite data networks using very small aperture terminals-part 1: Multi-access protocols. *Int. Journ. of Sat. Comm.*, 5:195-212.
- Wong, E. W. M. and Yum, T. S. (1991) A controlled multiaccess protocol for packet satellite communications. *IEEE Trans. on Commun.*, vol. 39 No. 7:1133-1140.
- Mohammed, Jahangir I. and Le-Ngoc, Tho (1994) Performance analysis of combined free/demand assignment multiple access CFDMA protocol

- for packet satellite communications. *Proceedings of ICC'94 Conference*, 869–873, New Orleans, 1994.
- Xu, Wenxin and Campbell, Graham (1993) A Distributed Queueing Random Access Protocol for a broadcast channel. *Proceedings SIGCOMM'93*, 270–278, Ithaca, N.Y., USA.
- Koperda, Frank and Lin, Bouchung and Collins, D. Jason (1995) A proposal to use XDQRAP for the IEEE 802.14. *Distribution IEEE 802.14 working group*, July 1995.
- Raychaudhuri, Dipankar (1985) Announced retransmission random access protocol. *IEEE Trans. on Commun.*, COM-33, No. 11:1183–1190.
- Capetanakis, John I. (1979) Tree algorithms for packet broadcast channels. *IEEE Trans. on Information Theory*, Vol. IT-25(No. 5):505–515.

# **A new HFC architecture using return path multiplexing**

*James C. Yee*

*Com21, Inc.*

*750 Tasman Drive, Milpitas, CA 95035, U.S.A.*

## **Abstract**

A Hybrid Fiber Coaxial (HFC) plant is typically configured in a tree topology and covers a large area with tens of thousands of House Holds Passed (HHP) and several return paths into the headend. During initial deployment, it is usually the case that the number of cable modem subscribers is small compared to the HHP, resulting in a small number of modems spread out over the return paths. To operate more efficiently, the return paths should be combined to reduce the port requirement at the headend.

A simple upstream RF combiner can be used to merge separate return paths, but that would also funnel the noise of the separate paths and degrade performance. Instead, what is needed is an upstream aggregation device that will multiplex the paths without aggregating the noise. To do so, such a device needs to operate with the HFC's MAC (media access control) layer and employ multiplexing discipline that incurs limited impact on performance under varying modem distribution scenarios. In addition, this device must be simple and transparent to the rest of the HFC system.

In this paper, we describe through analysis and simulation how such a return path multiplexing device is possible, and how it impacts the HFC network architecture and upstream performance.

## **Keywords**

Hybrid Fiber Coaxial, Cable Modem, Return Path Multiplexing

## 1 INTRODUCTION

Depending on the topology of the HFC plant, the distribution of users, and the node size, the number of return paths into the headend of a HFC plant may vary. For example, a HFC plant with 64 000 House Holds Passed (HHP) will result in 32 separate returns assuming a node size of 2000. Further recombination of these returns in each node using Fabre-Perot or Direct Feed-Back lasers may reduce the return port requirement by a factor of 4, yielding a realistic upstream port requirement of 8. A HFC plant with these components is illustrated in Figure 1.

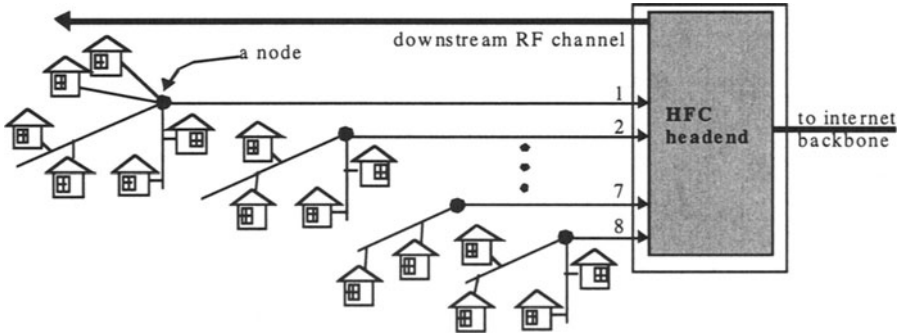


Figure 1: HFC access network with 8 return paths

The number of active subscribers on each trunk is likely to be small in initial deployment, perhaps only 1%. Such typical sparse deployment means each return path only services a few modems. Nevertheless, each return path normally requires a port in the headend. Since the cost of headend equipment increases with the number of ports, this results in an inefficient utilization of headend resources.

To increase efficiency, we need to concentrate the modems into fewer ports. To do so, we can either redesign the HFC plant topology or we can recombine the return trunks. The first solution is too costly and will backfire when more cable modem subscribers join. A simple upstream RF combiner can be used to combine the return paths, but that would also raise the noise floor at the headend ports due to a phenomenon called noise funneling. Noise funneling can be catastrophic to the performance of the HFC access network.

Instead, what is needed is an upstream aggregation device that will recombine the paths without aggregating the noise. To do so, such a device needs to operate with the HFC's MAC (media access control) layer and employ a multiplexing policy that incurs limited impact on performance under varying modem distribution scenarios. The complexity of this device must be low to realize the cost savings. In addition, like a passive RF combiner, this device must be transparent to the rest of the system. In this paper, we call such a device the Return

Multiplexer (RMUX), and illustrated in Figure 2 how it fits into a HFC access network.

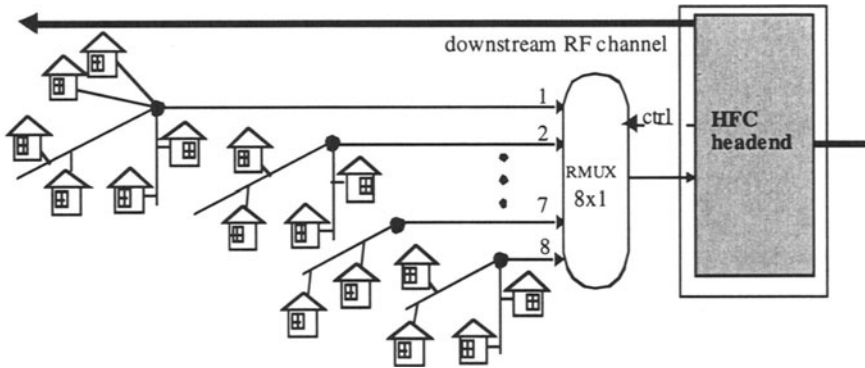


Figure 2: Reduced port requirement using a 8x1 RMUX

As we will describe in the next section, the RMUX introduces a new architecture for HFC systems. There are of course certain trade-offs and limitations in its use, but properly configured, the RMUX can dramatically reduce the headend port requirement with minimal performance degradation. In the remainder of this paper, we examine through simple analysis and detailed simulation how the RMUX performs under various configurations and load conditions.

## 2 SYSTEM DESCRIPTION

As described in the UPSTREAMS HFC MAC/PHY protocol specification (Laubach, 1997), upstream data in many deployed HFC access network is transported in fixed sized TDMA slots carrying ATM cell payloads, and the allocation of the slots is centrally scheduled by the headend. The RMUX, as a cell-level multiplexer, can therefore be controlled by the headend to open and close the corresponding path during the appropriate slots. This can be done without any loss of data cells. This is illustrated in Figure 3, where the RMUX will serve the incoming data cells in the port sequence of {1,2,4,3}. In this way, the RMUX isolates each return path from the RF noise of other paths.

Typical HFC protocol employs a request-grant mechanism for modems to acquire upstream data slots during preallocated phases of the protocol's operation. During the request phase, a fixed number of slots are made available for modems to send requests upstream. Such requests are sent according to a multiple access protocol with a collision resolution mechanism similar to that

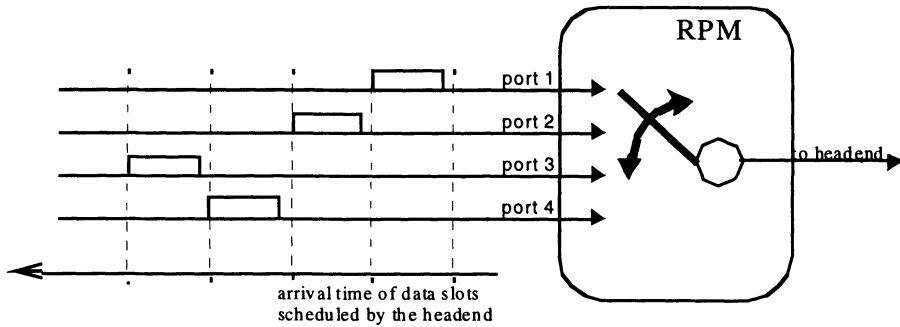


Figure 3: switched forwarding of data in a 4x1 RMUX

of 802.3 Ethernet. However, the headend has no a priori knowledge of when or on which return path the next request will arrive. Consequently, such a cell may arrive at a port not currently listened to by the RMUX. A request cell that is not serviced by the RMUX is lost, and is referred to as a blocked cell.

From the viewpoint of the transmitting cable modem, there is no difference between a blocked request cell and a request cell lost to collision. Therefore, the RMUX can be viewed as an additional contention mechanism during the request phase. To optimize the performance of the RMUX is to minimize the number of blocked cells, and to do so in a fair manner across the input ports of the RMUX.

The manner in which cells are blocked at the RMUX depends on two main factors. One is the service discipline used in servicing the input ports of the RMUX during the request phase. The other is the load distribution across the input ports of the RMUX. We next discuss these two factors.

## 2.1 RMUX Service Discipline

The RMUX service discipline dictates which input port to service and at which time. The simplest form of service discipline is the round-robin, and there are several variations.

- Round-Robin: by serving each input port in a cyclic manner and serve a constant number of arrivals during each visit, round-robin is a fair policy.
- Grouped Round-Robin: if certain ports can be identified as lightly loaded or have high SNR and can afford to withstand some noise aggregation, groups of input ports can be serviced at the same time, thereby reducing the latency between the sojourn time between visits. Again, a fixed number of slots is served during each visit.

Aside from round-robin, we can also vary the order of service of the input ports.



- **Load Dependent Acyclic:** the load on each input port can be used to determine how many consecutive slots are serviced each time and in what order the inputs are serviced.

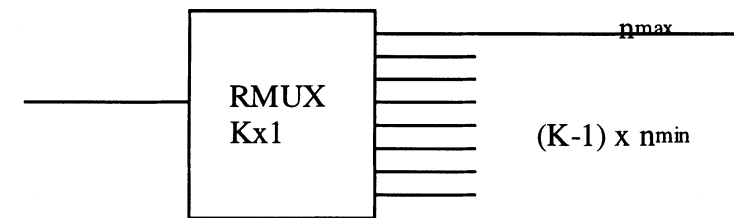
However, such acyclic policies are more complex to implement. We propose some in this paper that should be relatively easy to implement, but no performance results are available for them at this time.

## 2.2 RMUX Input Load Distribution

Assume the RMUX employs a round-robin policy. With the total number of modems fixed, the worst performance should occur when if all upstream traffic is grouped onto one input port. Reasoning similarly, the best performance will occur under an evenly distributed load across the input ports.

The probability of arrival at the ports can be used to quantify the load distribution across the input ports. However, such characterization is not very useful to the HFC access network architect, since such fine grained information is typically not available. A more practical way to quantify the load distribution is to use the number of active modems on each port. If the HFC system used is capable of providing Quality of Service (QoS) levels delineated by minimum and maximum rates, like that of Com21's system, then the subscribed minimum rates together with the number of modems can together be used to quantify the load on each port. For simplicity, we assume the QoS of all modems, if applicable, to be the same.

Let the number of input ports be  $K$ , we define the parameter Index of Symmetry (IOS) as illustrated in Figure 4 to better quantify the symmetry of the input port loading.



$N$  = total # of cable modems

$$= n_{max} + (K-1) \times n_{min}$$

$$IOS \text{ (Index of Symmetry)} = n_{min}/n_{max}$$

Figure 4: Illustrate Index of Symmetry

Table 1 Example Index of Symmetry calculations

$K$	$N$	$n_{min}$	$n_{max}$	$IOS$
8	128	16	16	1.0
8	128	0	128	0.0
8	128	14	30	0.467
4	64	14	22	0.636

Using this definition:

$$IOS = n_{min}/n_{max} \quad (1)$$

For the type of load distribution shown in Figure 4, we have

$$n_{max} = N/[(K-1)IOS + 1] \quad (2)$$

which can be conveniently used in network planning.

An IOS can be similarly defined for a system with bimodal load. That is, if the ports are partitioned into two group with group 1 having  $n_{max}$  cable modems and group 2 having  $n_{min}$  cable modems, where  $n_{max} \geq n_{min}$ , we can define the IOS to be  $n_{min}/n_{max}$ . Through simulation, we have found that regardless of how nodes are distributed, the IOS definition in (1) allows us to establish at least approximate regions of operation that gives desirable performance.

### 2.3 Implications

The above discussion tells us that there are many design and configuration alternatives associated with the RMUX. There are many extensions and variations of service disciplines based on the ones proposed. For example, a load dependent round robin policy can vary the number of slots served depending on the number of modems active on each port. An optimal service discipline that delivers zero blocking under all load distributions may exist, but it will require accurate estimates of traffic pattern of all modems and coordinate closely with the headend scheduler. The complexity of the resulting RMUX will likely be prohibitive. We will demonstrate in the following that a simple RMUX can deliver quite decent performance.

Even when the choice of service discipline and load distribution is not optimized to ensure no blocking, we must keep in mind of the alternatives facing the HFC

access network architect. If a RF combiner is used to aggregate 8 return paths, the signal to noise ratio (SNR) may decrease by as much as 9 dB. Even with the SNR of each return path at a favorable 20 dB or better, the impact on system performance is significant.

### 3 ANALYSIS

The RMUX can be modeled as a simple polling system where arrivals and service are slotted with slot duration  $D$ . Different from a typical polling system, however, the buffer size is zero and the switching time depends on the service time. We denote the  $n$ th time the server visits buffer  $i$  by  $t_i(n)$ , and the number of arrivals during this  $n$ th visit is  $a_i(n)$ . The maximum number of slots serviced during that visit is denoted by  $s_i(n)$ , and the switching time from buffer  $i$  to the next buffer after  $t_i(n)$  is denoted by  $w_i(n)$ . The value of  $w_i(n)$  is given by

$$w_i(n) = D(s_i(n) - a_i(n)). \quad (3)$$

In Figure 5 is shown such a polling system with  $K$  buffers. The load on port  $i$  is represented by the arrival process  $x_i$  into buffer  $i$ . Each arrival process  $x_i$  describes the slotted arrival of requests into port  $i$ . The process  $x_i$  needs to capture the behavior of the aggregated traffic generated by the modems connected to port  $i$ , as modulated by the Ethernet like contention resolution algorithm.

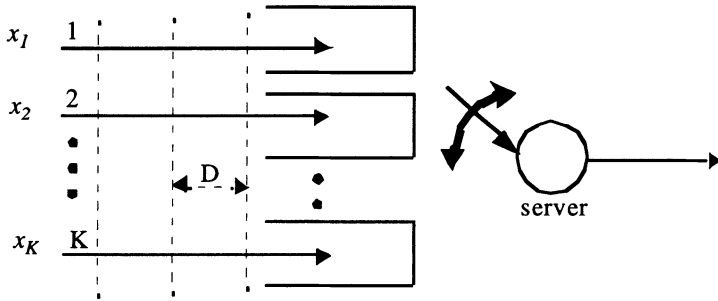


Figure 5: polling system model for the RMUX

There are existing analytical results (Takagi, 1985) which can be extended to describe the behavior for this polling system, but under unrealistic stochastic assumptions. For example, one possible metric for evaluating the performance of the RMUX is the probability that an arrival to buffer  $i$  will occur during the interval  $[t_i(n), t_i(n) + Ds_i(n)]$ , for some  $n$ . But such a metric requires a valid traffic model describing the arrival process, which is difficult to obtain.

For now, we only present some first order approximate analysis for the proposed service disciplines to illustrate the general behavior of the service disciplines. The more detailed evaluation of the system is left to simulation.

### 3.1 Round-Robin Service

We model a RMUX with round-robin service using a polling system with a cyclic server, where after buffer  $i$  is visited, the server moves on to buffer  $(i+1) \bmod K$ . We also require that  $s_i(n) = 1$ , for all  $i, n$ .

This is a system with symmetric load, so the probability that an arriving customer will not be blocked is  $1/K$ . Fortunately, this does not translate automatically into a scaling of throughput by  $1/K$ . Let the normalized occupation of upstream slots, or load, of a system without the RMUX be  $\lambda$ , where  $\lambda = 1$  implies that every upstream slot is occupied. Assuming traffic is divided evenly among the  $K$  ports, then the load on each port  $i$ , denoted by  $L_i$ , is  $\lambda/K$ . However, this does not mean only  $\lambda/K$  of the slots on each port will be occupied. With less contention on a path with fewer modems, the ratio of slots occupied is scaled up by a factor  $\alpha$ ,  $1 \leq \alpha \leq K/\lambda$ , as illustrated in the shaded region of Figure 6 (a). So scaled, the portion of slots occupied per port is  $P = \alpha\lambda/K$ .

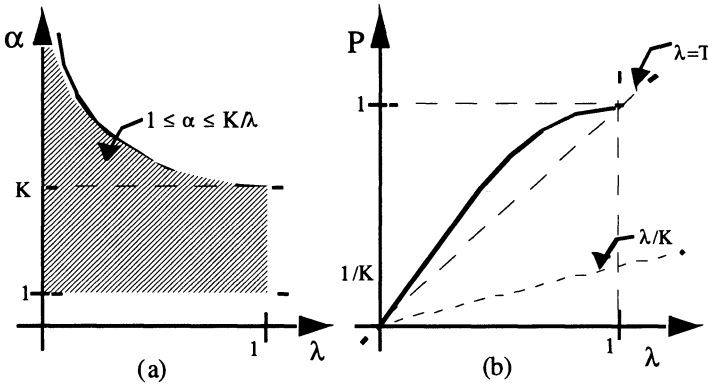


Figure 6: illustrate effects of traffic divided into  $K$  ports

The exact value of  $\alpha$  depends on many details of the binary exponential back-off algorithm used in the contention resolution algorithm, but we can interpret the boundary cases intuitively. In a highly loaded system with large  $\lambda$ , we have modems that failed to acquire requests slots previously utilizing the newly available request cell slots in the now lightly loaded return path, which results in  $\alpha \cong K/\lambda$ , and  $P = \alpha\lambda/K \cong 1$ . Only in a very lightly loaded system do we have  $\alpha = 1$ , where no addition request cells are generated. In this case, with  $\lambda = \varepsilon \cong 0$ , we can write  $P = \alpha\lambda/K \cong \varepsilon = \lambda$ . The resulting  $P$  is plotted in Figure 6 (b), where we see

qualitatively how the portion of slots occupied is increased from  $\lambda/K$  as a function of  $\lambda$ .

The round-robin RMUX scales the throughput of each port by allowing only  $1/K$ th of the slots through, which gives us a throughput of  $\alpha\lambda/K^2$ . Combining the  $K$  paths, the aggregate throughput is then  $T := \alpha\lambda/K$ . Using the value of  $\alpha$  reasoned above, we can plot the approximate throughput  $T$  as a function of  $\lambda$  in Figure 7 (a).

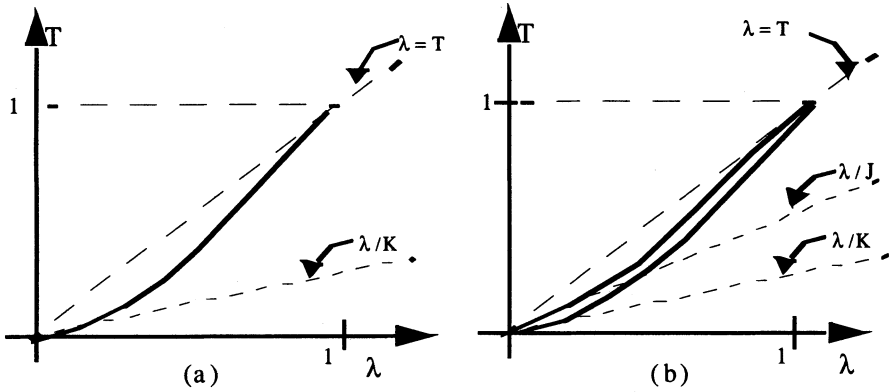


Figure 7: illustrate approximate analysis of RMUX throughput

### 3.2 Grouped Round-Robin Service

We model a RMUX with a grouped round-robin service using a polling system with a cyclic server, a reduced number of buffers, and aggregated arrival processes. Let  $J$  be the number of aggregated arrival processes, where  $J < K$ . As before, we have  $s_i(n) = 1$ , for all  $i, n$ .

For simplicity, we examine the case where  $J = K/2$ , where adjacent ports are grouped into pairs. With the number of input ports reduced by  $1/2$ , the probability that an arriving request cell will not be blocked is now  $2/K$ . Following a similar line of reasoning as above, we have  $1 \leq \alpha \leq K/2\lambda$ , and the throughput  $T$  is bounded below by:  $\lambda/J$ . Since the increased aggregated load on each port will push  $\alpha$  towards  $K/2\lambda$ , we argue that the throughput  $T$  will be approximately as shown by the top bold line in Figure 7 (b).

We can therefore expect that grouping of ports, when allowed under noise and load conditions, will result in better performance.

### 3.3 Load Dependent Acyclic Service

We model a RMUX with a load dependent acyclic server using a deterministic fluid approximation polling system model with a rate dependent service discipline.

Instead of fixing  $s_i(n)$  based on a fixed loading of the port, the service discipline now adjusts  $s_i(n)$  dynamically as a function of time and the load on each port.

Extending existing scheduling results (Clear-A-Fraction policy from (Perkins)), we propose the following policy where  $s_i(n)$  is based on estimated per port load.

The policy we propose is called the Highest-Load-First (HLF) policy. Let  $t_m$  be the time at which the server visits the  $m$ th buffer. Let  $e_i(m)$  be the estimated number of arrivals into buffer  $i$  during  $[t_m, t_m + s_i(\max)]$ , where  $s_i(\max)$  is a predetermined maximum number of slots each port can be allocated at any time. Such a value can be easily selected based on maximum latency of a HFC system, as well as based on minimum allocations to each port. With HLF, after the server has finished serving a buffer, the next buffer to be visited is buffer  $j$ , where  $j = \max\{i | e_i(n)\}$ . The number of slots serviced at port  $i$  is then set to  $s_i(n) = \min\{e_i(n), s_i(\max)\}$ .

Earlier work (Perkins) has shown that such a policy is complete, that is, a set of  $s_i(\max)$ ,  $i = 1, \dots, K$ , can be selected to ensure that every buffer will be visited in a given time period. A more generalized version of the HLF is the Fractional-Load (FL) policy, where the next buffer  $i$  is selected based on whether  $e_i(n) > \varepsilon \sum_j e_j(n)$ , where  $0 < \varepsilon < 1$ , is satisfied. Assuming the estimated load equals the actual load, these scheduling policies have been shown to be stable under different initial conditions. The key potential advantage of these policies is that they are relatively easy to implement and have parameters that can be tuned to optimize performance.

#### 4 SIMULATION MODEL

The analytical models above do not describe the behavior of the system quantitatively, nor do they capture the performance of the system under realistic traffic. By realistic traffic, we mean the traffic generated by the typical cable modem user. That traffic is TCP/IP based, and is mainly generated by World Wide Web (WWW) and FTP client-server application. Such traffic introduce yet another layer of flow control and also is influenced by human behavior not easily captured by analysis. For these reasons, we rely mostly on simulation.

The simulator we use is a customized version of the *ns* simulator (Ns, 1998), a timed discrete event simulator. The simulator contains modules for simulating the UPSTREAMS HFC MAC protocol (Laubach, 1997), full TCP protocol behavior, and WWW browser client-server interactions as described in (Nichols, 1997). Our WWW client-server model, including file size distributions, is based on previous trace data (Cunha, 1995).

Using this version of *ns*, we simulated the system using the following parameters:

- Number of users/modems: Several user population sizes (ranging from 64 to 2000) are used.

- HFC network: For simplicity, we assume a single upstream receive port at the headend and a single downstream channel. Depending on the RMUX configuration, four or eight upstream return paths feed into the single RMUX.
- Quality of service: the HFC system simulated allows maximum and minimum rates to be specified for both up and downstream traffic.
- Servers: For simplicity, we assume there is no limit to the number of concurrent WWW sessions using the HTTP protocol at the servers.

In Figure 8, we illustrate the major components in our simulation model. Important to note is that this is a detailed model that takes into account the behavior of the TCP protocol, but also the interactions between the ATM protocol (e.g., Segmentation and Reassembly - SAR agent), the UPSTREAMS protocol (Laubach, 1997), TCP, and higher layer applications (e.g., WWW/FTP server and clients).

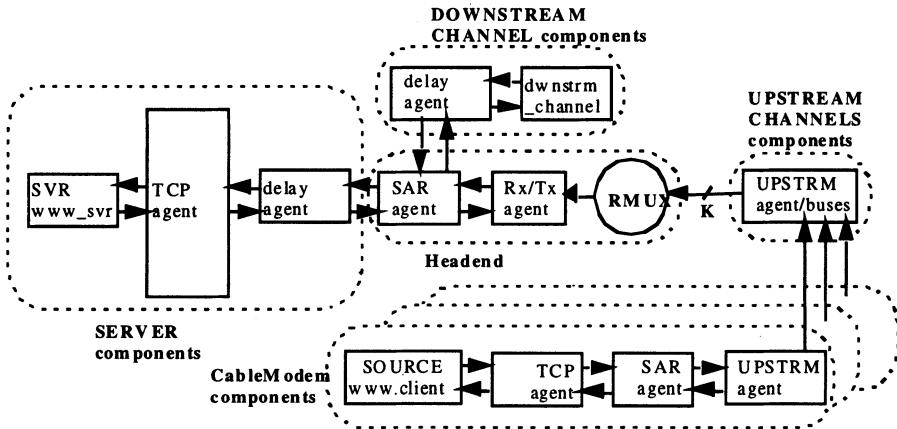


Figure 8: components in the simulation model

During each WWW session between a client and the headend server, a sequence of short and long packets containing TCP SYN and SYN+ACK, URL, and document pages is exchanged. As described in more detail in (Nichols, 1997), each such WWW browser exchange is currently configured with the following parameters:

- TCP window size of  $6 \times 1460 = 8760$  bytes
- Packet size of 64 bytes.
- URL size = 3 cells
- Page size distributed according to pareto distribution up to 1.5MB
- Each page contain 4 in-line documents (e.g., images) whose size is distributed according to the pareto distribution up to 0.25MB
- Initial delay of 0.

The resulting average page size is around 10KB, while the in-line document size is around 5KB.

The duration of a typical simulation run is between 150 and 600 seconds in simulated time. This translates into real time depending on the number of events generated, which is proportional to the number of cable modems, load, and collision frequency. Using an Ultra Sparc workstation, a moderately loaded simulation run with 512 cable modems and 300 simulated seconds takes around 3 hours to complete, while a simulation run with 16 cable modems takes about 15 minutes. To eliminate the effects of transient behaviors, the initial 30 seconds of all simulations runs are excluded from the final tallied results.

## 5 SIMULATION RESULTS

Using the simulated WWW traffic above, we simulated the performance of a HFC system with a RMUX.

### *Performance metrics*

The performance metric used are:

- Latency: median and 90 percentile end-to-end delay for each modem.
- Throughput: average throughput for each modem.

Statistics about cell level, packet level, and page level performance are recorded, though only cell and packet level statistics are presented here.

### **5.1 Effects of Load Distribution**

In Figure 9, we have plotted the upstream latency comparisons between varying number of cable modems. The number of modems was varied from 64 to 2,000, and clearly independent of the number of modems, the index of symmetry (IOS) as defined in equation (1) was the dominant factor in the latency performance.



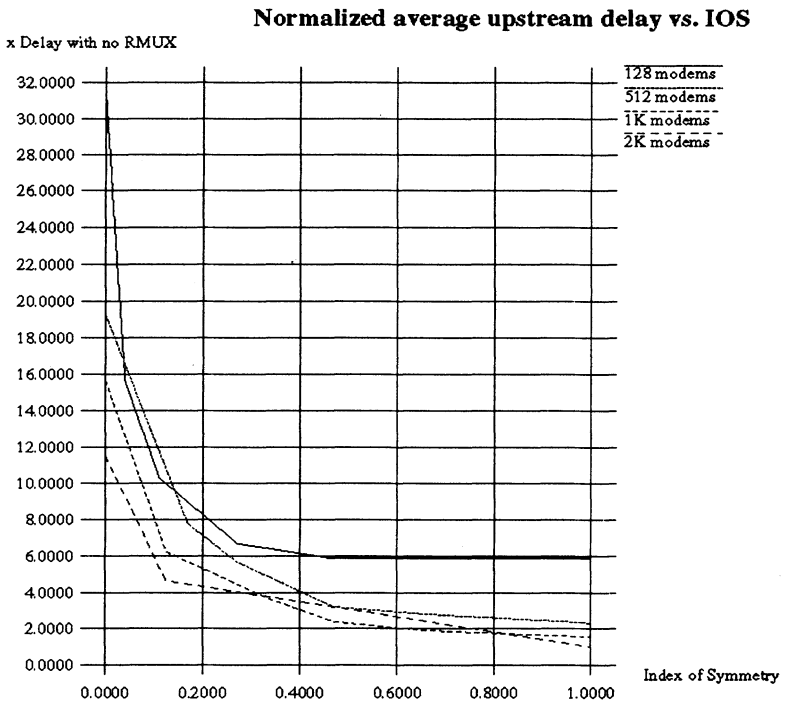


Figure 8: upstream latency as a function of Index of Symmetry

From Figure 9, we see that increased IOS brings about dramatic improvement in the delay behavior. The curves also flatten out to around 2 for an IOS value of approximately greater than 0.4. Using equation (3), this threshold of 0.4 implies that one should place no more than about  $1/4$  ( $N/3.8$ , to be exact) of the cable modems on one input port in a  $8 \times 1$  configured RMUX, or no more than  $N/2.2$  cable modems on one input port of a  $4 \times 1$  RMUX.

What this implies for throughput is a more complicated question to answer. This is because how increased upstream latency manifests itself in TCP throughput degradation depends on many factors. In (Cohen, 1998), the effects of upstream delay and error on TCP throughput was studied in detail. Although the study did

not take into account the behavior of the HFC MAC, it does illustrate the sensitivity of TCP throughput to the HFC upstream integrity.

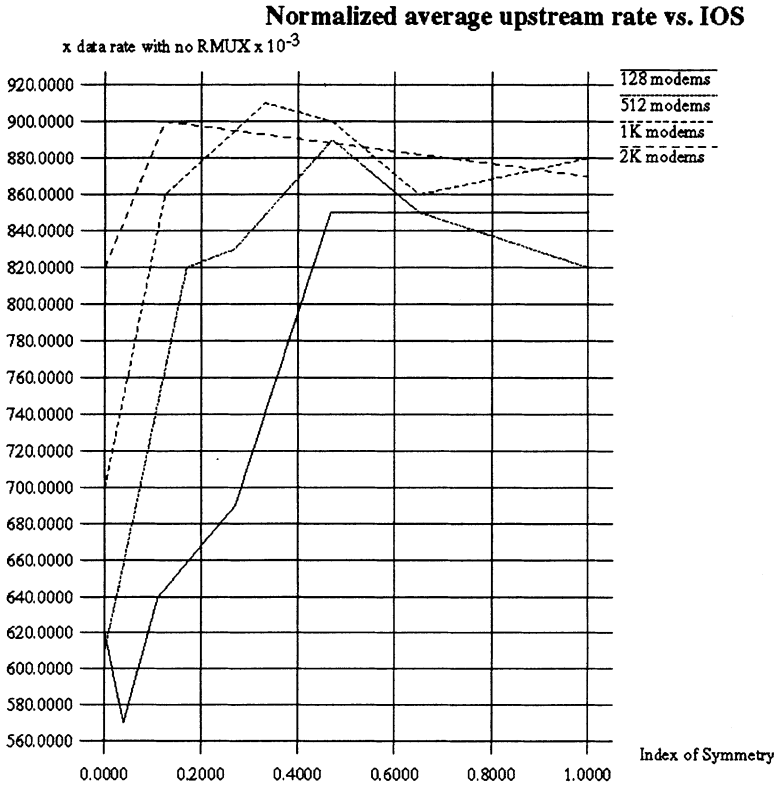


Figure 10: upstream throughput as a function of Index of Symmetry

In Figure 10, we see that there is a knee in the curves at an IOS value of 0.2 - 0.4, and the throughput of highly loaded systems are much closer to the throughput of a system without the RMUX. This was predicted by the simple analysis illustrated in Figure 7. It is not clear why the throughput appears to decrease as the

IOS approaches 1, but it seems that the decrease is accompanied with a decrease in the throughput variance.

## 5.2 Effects of Service Discipline

We present here a comparison of Round-Robin vs. Grouped Round Robin.

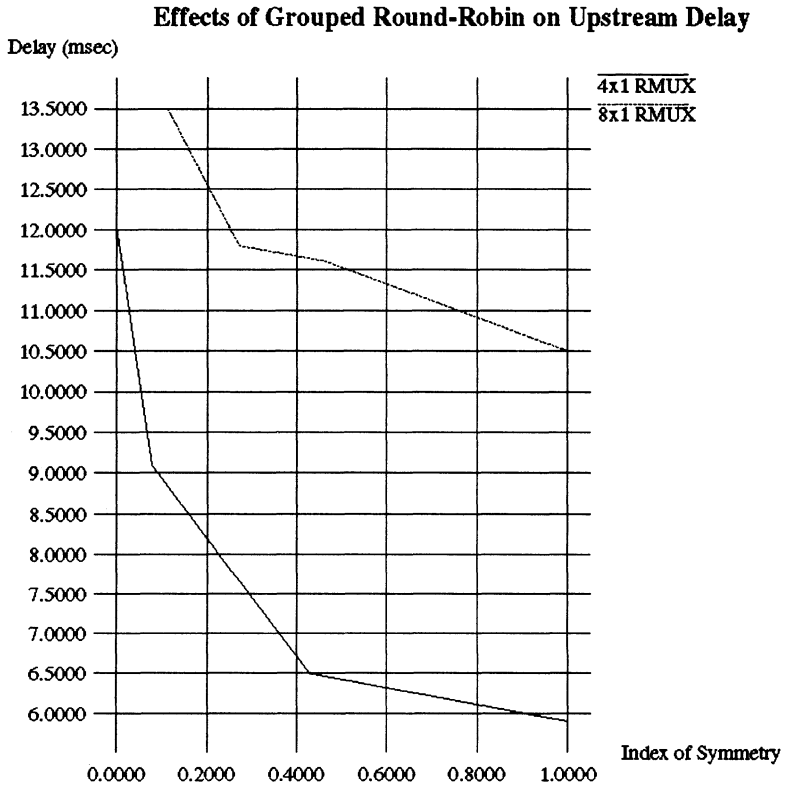


Figure 11: latency improvements from grouped round-robin

In Figure 11, we plotted two sets of simulated latency results from WWW client-server session for a fixed number of modems. We see that by grouping lightly loaded ports into pairs, the resulting latency of TCP transfers is significantly improved. In the  $K=4$  case, the upstream delay varied from 6 to 9 msec. For  $K=8$ , we see that the latency increased to the range of 10 to 13.5 msec, a substantial increase.

This implies that the HFC system architect should identify lightly loaded ports and configure the RMUX to serve those ports concurrently. Doing so will greatly lower the average upstream latency of the entire system. This result also tells us that selecting the smallest  $K$  possible when configuring a RMUX will improve the system performance.

## 6 CONCLUSIONS

Focusing on the scheduling disciplines of a RMUX and how its performance depends on the load distribution of the upstream traffic, we have illustrated through simple analytical and simulation results that Return Path Multiplexing is a useful way of reducing headend port requirements (by a factor of  $K$ ) in a HFC plant without significant performance degradation. Properly configured, simulation results show that greater than 80% throughput can be achieved. The RMUX's multiplex service discipline only operates during the request phase on top of the contention resolution mechanism of the media access protocol (MAC), and does not affect the operation of the MAC during the data transfer phase.

The simulation results presented in this paper used realistic simulated WWW traffic sources, and validated some of the simple analytical results. However, it is clear that a more detailed simulation study of the RMUX behavior is needed. Specifically, effects of return path multiplexing on TCP level downstream throughput and latency needs to be studied.

We see that a simple cell level switch like the RMUX dramatically changes the architecture and provisioning policy of a cable modem system. Rather than having to accept a given physical layer topology and plant RF characteristic, the RMUX allows the cable operator to change a given topology without detrimental performance consequences typical of current passive RF combination methods. Whereas capacity planning and resource allocation of the HFC plant used to be issues constrained by the given RF condition of the return paths, now a device like the RMUX allows the cable operator to dynamically manage or isolate noisy return paths.

However, we have also seen that it has an operating region that the HFC architect must be made aware of. Although the operating region identified in these results is large and can be easily engineered, an RMUX with an overly asymmetric load, as illustrated by the simulation results, will see performance degradation. In

order to properly configure the RMUX, a HFC system that allows the operator to provision the bandwidth allocation of each modem is required. This highlights the need for the ability to provision and enforce QoS in a HFC system.

There are many aspect of the RMUX's operation which we have not touched on. As with anything in the HFC plant, its proper performance depends primarily on the RF condition of the plant. Indeed, with favorable SNR in the HFC plant, it may be acceptable to service all the input ports of the RMUX concurrently during the request phase. This will allow the RMUX to operate without blocking, and simply let the contention resolution of the HFC protocol operate normally. Note that since the RMUX does not funnel noise during the data transport phase of the protocol's operation, the resulting performance will be substantially better than using a simple passive RF combiner. The precise economic impact of a device like the RMUX is an important topic worthy of study, and investigations into other methods of achieving better RMUX performance are also ongoing.

### *Disclaimer*

The results presented in this paper do not necessarily represent the performance or design of Com21's commercial product that enables return path multiplexing.

### *Acknowledgments*

The idea of return path multiplexing originated not with the author. Instead, it came from colleagues at Com21 driven by HFC system deployment experience. A tremendous amount of effort by the whole Com21 team went into the research and development of the commercial product based on the idea. They are the ones who made this paper possible.

## 7 REFERENCES

- Cohen R. and Ramanathan S. (1998) TCP for High Performance in Hybrid Fiber Coaxial Broad-Band Access Networks, *IEEE/ACM Transactions on Networking*, Vol 6, No. 1.
- Cunha C.R. et al, Characteristics of WWW Client-based traces, Boston University Computer Science Technical Report, BU-CS-95-010, July 18, 1995.
- Laubach, M. The UPSTREAMS Protocol for HFC Networks, Version 1.09 970623, SCTE-DSS-97-xx, (ed. M. Laubach, Com21, Inc.), June 23, 1997.
- Ns (1998), University of California at Berkeley/Lawrence Berkeley National Laboratory/VINT Network Simulator, <http://www-mash.cs.berkeley.edu/ns/>
- Nichols K.M. and Laubach M., Tiers of Service for Data Access in a HFC Architecture, *Proceedings of SCTE Convergence Conference*, January, 1997.
- Perkins J.R. and Kumar P. (1989) Stable, Distributed, Real-Time Scheduling of Flexible Manufacturing/Assembly/Disassembly Systems, *IEEE Transactions on Automatic Control*, Vol 34, No. 2.

Takagi H. (1986) Analysis of Polling Systems, The MIT Press, Cambridge, Massachusetts.

## 8 BIOGRAPHY

James C. Yee is a Performance Engineer at Com21, Inc., a company in Milpitas, CA, USA that makes cable modem head-end equipment, cable modems, network management software, and noise-containment technologies. He received his B.S. from Columbia University in 1987, M.S. and Ph.D. in Electrical Engineering and Computer Sciences from U.C. Berkeley in 1988 and 1994, respectively. He is also an adjunct faculty at the Computer Engineering Department of Santa Clara University.

His research interests include network performance modeling and analysis, real-time scheduling, ATM, and IP services. Previously, he had worked as a networking consultant, at Pacific Telesis Group, and at Philips Research.

## **Part Two**

---

# **Multimedia Multicast**

# **End-to-End Reliable Multicast Transport Protocol Adaptation for Floor Control and Other Conference Control Functions Requirements**

*Nadia Kausar, Jon Crowcroft  
Department of Computer Science  
University College London  
Gower Street, London WC1E 6BT, UK  
Phone +44171 504 4433  
Fax +44171 387 1397  
n.kausar@cs.ucl.ac.uk, jon@cs.ucl.ac.uk*

## **Abstract**

In order to provide guaranteed QoS multiparty collaborative multimedia applications require reliable transmission of data. The multimedia applications can vary from distributed games, shared whiteboard to interactive video conferencing. These applications often involve a large number of participants and are interactive in nature with participants dynamically joining and leaving the applications[Sudan95]. In order to provide many-to-many interaction when the number of participants is large IP multicast is a very good option for communication. IP multicast provides scalability and efficient routing but does not provide the reliability these multimedia applications may require. Though a lot of research has been done on reliable multicast transport protocol, it really seems that the only way of doing a reliable multicast is to build it for a given purpose like conference control in multimedia conferencing.

This paper compares some of the available multicast transport protocols and analyses the most suitable features and functionalities provided by these protocols for a facet of conference control, floor control. The goal is to find or design a reliable multicast transport protocol which would scale to tens or hundreds of participants scattered across the Internet and deliver the control messages reliably.

## **Keywords**

**Floor control, reliable multicast, conference control**



## 1 INTRODUCTION

Conferences come in many shapes and sizes, but there are two models of conference control. These are known as Formal/tightly coupled conferencing and Informal/loosely coupled conferencing. Lightweight/informal sessions are multicast based multimedia conferences that lack explicit conference membership control and explicit conference control mechanisms. Typically a loosely coupled session consists of a number of many-to-many media streams supported using RTP and RTCP using IP multicast. Typically, the only conference control information that is provided during the course of a light-weight session is that distributed in the RTCP session information, i.e. an approximate membership list with some attributes per member.

On the contrary, tightly coupled conferences where the media streams are flowing from mainly one-to-many or one-to-one basis, requires an explicit conference control mechanism. In a model like that a user interface is provided where the chair can choose to give a floor to one of the participants, so one person can talk, take control of the shared whiteboard or use the video channel at a time.

The most conventional tightly coupled conferences are ITU based H.323[H.323] or T.120[T.120] standard conferencing which was initially designed to work over circuit switched networks like ISDN and the loosely coupled conferences are Mbone[MboneFAQ] based which are designed for IP multicast. Some features of the tightly coupled conferences like floor control have only recently been designed to work on IP with TCP over it or use UDP for other type of data. Therefore, the most suitable reliable IP multicast for tightly coupled conferences is a recent issue.

IP Multicast provides a service model by which a group of senders and receivers can exchange data without the senders needing to know who the receivers are\*, or the receivers needing to know in advance who the senders are. Hosts that have joined a multicast group will receive packets sent to that group. Therefore, this service model can lead to applications which will scale to hundreds/thousands or more receivers. Although, because of the limited bandwidth most applications like videoconferencing will deploy floor control to limit traffic from the group to a small number of concurrent sources.

In order to support floor control either for a tightly coupled session (where reliability and ordering of the messages may get the highest priorities) or a loosely coupled session (where congestion control or retransmission strategy may be more complex and more critical than strict ordering), certain characteristics from a multicast protocol are required. The requirements for conference control from a transport protocol are:

1. Reliability and loss detection
2. Retransmission strategy, queue management
3. Scalability - source to many receivers, many sources to many receivers etc

\* Unless a higher level agreement has been done.

### Ordering

4. Scope of membership
5. Congestion control
6. Integrated security

A lot of research is being done on reliable multicast transport protocols. This paper looks at some of the available protocols like SRM, MTP/SO, RMTP, RLC and PGM and compares them against the requirements of single facet of conference control Floor Control. The reason for choosing these particular protocol is that they provide a lot of the functionalities required by a conference control mechanism. However, there may well be other protocols available now or may well be in the design phase which may serve the same purposes.

Section 2.0 is the background of some of the available reliable multicast protocols, section 3 analyses floor control and its requirements in general, the following section looks at the limitation of floor control, section 6.0 highlights the limitations of some of these multicast transport protocols in the light of floor control requirements. The last sections (7.0 and 8.0) describes an ideal reliable IP multicast with certain characteristics and which of the available protocols provide some of these functionalities.

## 2 BACKGROUND

Loss detection and retransmission strategy are two important aspects in the design of any reliable protocol. In a reliable transport protocol a recipient can (within bounded time) find out when it is failing or being partitioned from active senders. A sender is assured (with sufficient probability) that all its messages reach within bounded time.

In a traditional point-to-point reliable protocol such as TCP, positive acknowledgements are used to detect loss and the sender is responsible for retransmission of the packet. Using TCP one can provide HTTP Web traffic, FTP file transfers, and e-mail. All TCP traffic is unicast, that is it has one source and one destination. The nature of data can be either bulk data transfer where all data is sent one way and then the sender waits for a response or interactive where as soon as each data unit is sent acknowledgement has to be returned. The transmitter sends out a window's worth of data before requiring an acknowledgement.

It is harder to transfer data "reliably" from source(s) to R receivers (where R can be 10's to 100,000 or more), because multicast protocols interact with multiple parties simultaneously and so involve a higher number of links. Therefore, the likelihood is greater that some of the paths in the source's multicast tree are unstable at any time. In addition, the instability in any portion of the multicast tree may affect many members of the group because of the collaborative adaptive algorithms used[Floyd98]. In particular, it is difficult to build a generic reliable transport protocol for multicast, much as TCP is a generic transport protocol for Unicast. Reliable multicast is a case where "one size fits all" does not work at all.

Applications often have very different reliability and latency requirements, state management styles, error recovery and group management mechanisms. A reliable multicast transport protocol that meets the worst-case requirements is unlikely to be efficient and scalable for many application requirements[Zhang97].

In a teleconferencing environment, a desirable robustness property is the ability to continue operating within partitions should the group become partitioned. Ultimately, the applications that use the multicast transport platform should be the ones to decide when the situation has deteriorated to a point where continuing is meaningless.

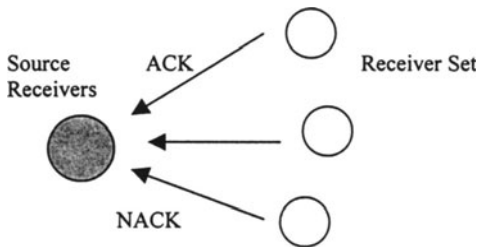


Diagram 1: A basic diagram of a sender initiated Protocol

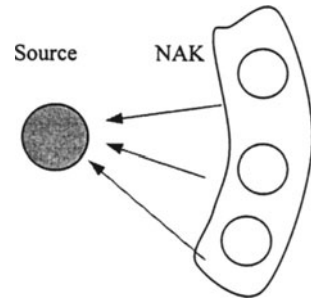


Diagram2: Receiver initiated Protocol

The design of a reliable IP multicast can be based on either a tree-based, a ring based or an ACK/NACK i.e. acknowledgement structure.

In the following subsection we provide an overview for some of the reliable multicast transport protocols:

## 2.1 SRM

Scalable reliable multicast (SRM) has been embedded into an Internet collaborative whiteboard application called wb. In SRM, whenever a receiver detects a packet loss, it multicasts a NACK packet to the entire group. Upon receiving the NACK packet, any member holding the desired packet can multicast it to the group. To avoid duplicate NACK and repair packets, a suppression algorithm is used in which a node sets a random timer before multicasting a NACK or repair packet. The messages specify a time-stamp used by the receivers to estimate the delay from the source, and the highest sequence number generated by the node as a source. SRM's implementation requires that every node stores all packets or that the application layer stores all relevant data.

One of the problems with SRM is that this algorithm will end up consuming a lot of bandwidth when there is little correlation of losses among receivers. For example, in a group of 1000 receivers, when only one receiver loses a packet, all 1000 receivers need to process the multicast NACK and repair packets. This

causes significant overhead. Also if one set of hosts in particular requires a packet, it is not desirable to multicast the packet to all the possible groups. One possible method of improving SRM's efficiency is to use localised recovery. The idea is to multicast NACKs and repairs locally to a limited area instead of to the whole group. Using the TTL (Time to Live field in the IP packet header is one possible way to implement scope control.

## 2.2 MTP/SO

Multicast Transport protocol or MTP provides an atomic and reliable transmission of messages. MTP/SO provides global ordering where messages are assigned to different streams. Therefore the delay caused by global ordering (for example when a short message is preceded by a very long one) is eliminated. MTP/SO proposes self-organisation of the members of a group into local regions for addressing the NACK implosion problem. MTP/SO provides a rate controlled transmission of user data. There are three main groups of members within a group: co-ordinator, repeaters and normal members. To provide maximum throughput the co-ordinator can send and receive retransmission, whereas if it is a type of a member who is just 'listen only' capable, the only packet type they can send to the group is unreliable multicast datagrams.

The rate controlled transmission of user data is very useful for floor control. If only few users are capable of holding the floor then there is only little point of giving all the other 10,000 receivers the capability of asking for retransmission of floor request. Although a lot of the functionalities of this protocol can be used for conference control (which is discussed in the section) purposes, the implementation of MTP/SO is in very early stage yet.

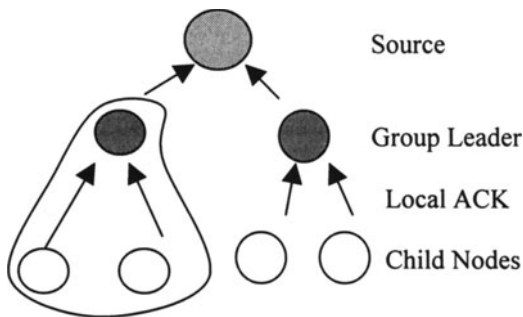
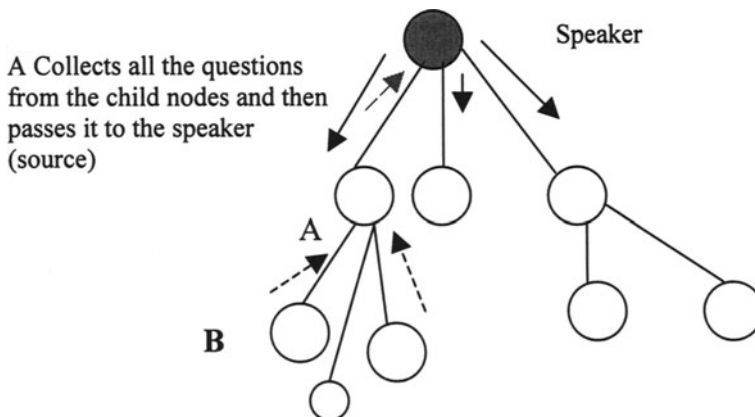
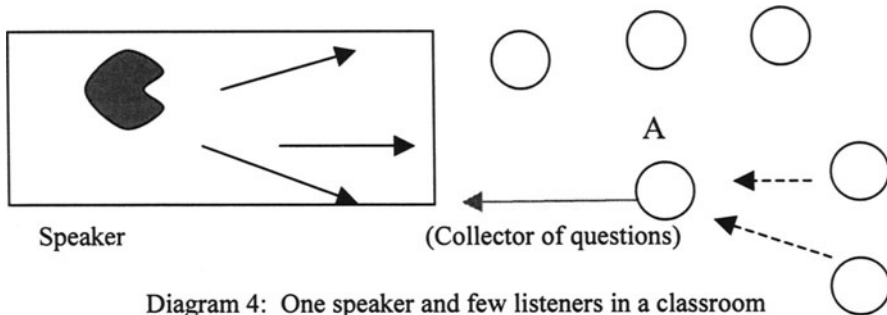


Diagram 3: A basic diagram of a tree based protocol

### 2.3 RMTP (Globalcast Communication)

Reliable Multicast Transport Protocol (RMTP) organises all the nodes into a tree structure. The receiving nodes are always at the bottom of the tree. Ideally the senders are at the top. The sender transmits messages using IP multicast, after a message is transmitted the sender will not release the memory until it receives a positive acknowledgement from the group. The receivers do not send acknowledgement directly to the top node(sender), but send hierarchical acknowledgements (HACKs). A receiver transmits a HACK to their parent in the tree structure. The parent gathers all HACKs from its children and sends a HACK to its parent node one step higher in the tree. The HACKs are propagated upward to the top of the tree and the sender is eventually notified. This design allows dissemination of messages to a large number of receivers without causing ACK implosion.

If there are lots of listeners and two or three speakers in a conference then this is a good architecture. As diagram below represents a floor control scenario in RMTP type of architecture.



## 2.4 RLC and RMDP

In Reliable Multicast data Distribution protocol (RMDP), the problem of insuring reliable data delivery to large groups, and adaptability to heterogeneous clients is solved by Forward Error Correction (FEC) technique based on erasure codes[Vicisano98].

The basic principle behind the use of erasure codes is that the original source data, in the form of a sequence of  $k$  packets, along with additional  $n$  redundant packets, are transmitted by the sender, and the redundant data can be used to recover lost source data at the receivers. A receiver can reconstruct the original source data once it receives a sufficient number of ( $k$  out of  $n$ ) packets. The main benefit of this approach is that different receivers can recover from different lost packets using the same redundant data. In principle, this idea can greatly reduce the number of retransmissions, as a single retransmission of redundant data can potentially benefit many receivers simultaneously.

In order to deal with *congestion control*, the ultimate problem of one-to-many data transfer protocols on top of the IP multicast, RLC (receiver driven layered congestion control) is proposed by the same authors. This mechanism is designed for a transmitter sending data to many receivers on the Mbone[Levine98]. In Unicast communications, the sender takes part to congestion control by changing its sending rate according to the congestion signal that it receives. In multicast communications, this approach would be problematic, since different groups of receivers with different requirements exist, and adapting to the need of one set of receivers will be unfair to the rest. The effect of congestion control is decided by the receivers. It gives receivers the possibility to modulate the receive rate by joining/leaving layers.

Though the above mechanisms are very good solution for bulk data transfer, it does not really satisfy the needs for floor control. For example, in floor control mechanism the identity of the participants are quite crucial. Combination of RLC + RMDP is not really appropriate for floor control purposes.

## 2.5 PGM

Pretty good multicast (PGM) is a reliable transport protocol for applications that require ordered, duplicate free, multicast data delivery from multiple sources to multiple receivers[speakman98]. When a receiver detects a missing packet, it repeatedly unicasts a NAK to the last-hop PGM network element on the distribution tree from the source. A receiver repeats this NAK until it receives a NAK confirmation (NCF) multicast to its group from that PGM network element. The network element repeatedly forwards the NAK to the upstream PGM network element on the reverse of the distribution path from the source of the original data packet until it also receives an NCF from that network element. Finally, the source itself receives and confirms the NAK by multicasting an NCF to the group.

PGM is not intended for use with applications that depend either upon acknowledged delivery to a known group of recipients, or upon total ordering amongst multiple sources. For floor control, these two functionalities are quite crucial, therefore PGM is not the best suited protocol for floor control. PGM is

better suited for applications in which members may join and leave at any time, and that are either insensitive to unrecoverable data packet loss or are prepared to resort to application recovery in the event.

## 2.6 Functional Criteria

The table below is a comparison of several multicast transport protocols based on functions that are relevant for floor control.

<i>Protocol</i>	<i>Reliability Semantics</i>	<i>Con- gestion Control</i>	<i>Participant structure</i>	<i>Knowledge of participant</i>	<i>ACK / NACKs/ Retrans- mission</i>	<i>Unit of delivery</i>
SRM	Reliable	No	Distributed	Via session messages	NAK, receiver reliable	1 ADU = app. Data unit
RMTP (BELL Labs)	Reliable	Yes	Hierarchy of regions, domain regions	Optional, may be known	Window of packets ACK/ HACK	N = window size
RLC + RMDP	Reliable	Yes	No	No	No ACK/ NAKs FEC for error recovery	K/N = depend on file size
PGM	Reliable	No	Local retrans- mitters	No	Bread crumb	1 packet
MTP/SO	Reliable, totally ordered, atomic delivery	Through different streams	Master  Repeater Consumer	Known	NAK (?)	?
NTE	Reliable	No	Distributed	Via Session packets	Trigg- ered NAKs with randomi- sation + FEC	1 ADU = 1 packet

### 3 FLOOR CONTROL AND ITS REQUIREMENTS

Floor control in CSCW is a metaphor for "assigning the floor to a speaker", which is not only applicable to voice channels, but more generally to any kind of sharable resource within conferencing and collaboration environments[Dommel95]. A floor is an individual temporary access or manipulation permission for a specific shared resource, e.g., a telepointer or voice-channel, allowing for concurrent and conflict-free resource access by several conferees.

For example, a floor requester in a meeting room would be a person who raises his/her hand up to ask a question. It is up to the chair to grant the floor to the requester. The session parameter entails the number of collaborators, and their role (chair, listener, a floor holder), determining their capabilities. Also, the interconnectivity (1-1, 1-to many, many-to-many), sharing distribution range(local, wide area), and link types (bi or unidirectional) are important too.

There are several types of floor control policy available for use by collaborative environments like explicit release, free floor, round-robin scheme etc.[Greenburg 91]. Whatever the scheme is, for applications to scale beyond a few participants, all communication must be multicast. Some research has been carried out to support Interactive collaboration application like TMTP[Sudan95] for data , STORM[Xu97] for audio and video and SRM[Floyd95] for wb. However, the nature of floor control is somehow different to these interactive applications. For example, the volume of data i.e. floor control messages are lot less than audio or video or whiteboard associated data, the timing of requesting/granting floor control can be very specific (for example, when the chair/speaker addresses the audience and asks for questions, a lot of listeners are going to request the floor but before that traffic may be lot less), ordering of data is more crucial factor than audio/video(for fairness, or applications like when customers are bidding for share) etc.

Typically traffic control for floor requests would be done in low level per source. An example of sudden flood of traffic would be "Flash Call" problem in POTS. Flash call would occur when a televoting system is taking place, where the viewers call a telephone number provided by a particular program, to give their opinion. The first method to avoid this sort of problem is the undeterministic approach, where after certain calls being taken by the network, users would hear an equipment engaged tone. This would stop the network being flooded by too many calls. Other approach is the deterministic approach, where the telephone company would be warned in a day advance, by the program organisers. So the telephone company can provide enough resources for that sort of service, and the cost would be higher.

On a data network, similar situation can take place too. There are certain traffic problems which would only apply to floor control and conference control type of applications. A reliable IP multicast protocol has to include certain features which would account for:



*Congestion control* - The volume of traffic will increase at certain points of time. The reliable IP multicast has to cope with sudden burst of traffic. Many sessions have precise starting times, when most of the members of a conference joining the session, or multimedia tools such as vat and vic can be programmed to join a session at the instant of its inception. This will cause a flood of traffic.

*Ordering* - To be fair to all the floor requesters the IP multicast has to have a mechanism for strict ordering. Let us consider if a receiver A requested for a floor who is 120 ms from the server/chairman. Receiver B requests for a floor who happens to be 100 ms away from the receiver/chairman after 10ms. Therefore, B's request will get to the server/chair before A's request, which is unfair for A. In this particular case the timing difference is so small that it may not really matter, but the difference can be in seconds rather than milliseconds.

*Reliability* - To provide good services, reliability and the retransmission strategy is quite important. Assume the scenario, where a floor request is multicast by a receiver A, receiver B didn't receive the message after time  $t$ . Receiver B now bids for the floor, without knowing the floor requester is Receiver A. Imagine there is a policy in this conference that if someone has requested a floor, the next person is not allowed to bid for the floor within next  $t'$  seconds. Now somehow in this scenario, someone has to inform the receiver B that receiver A has asked for the floor, and he may not request/being granted the floor. In protocols like TMTP the domain manager retransmit the data, whereas in SRM the nearest receiver to B will transmit the data.

*Member Classes* - There can be different types of members in a conference. As discussed in section 2.2, the rate controlled transmission of user data is very useful for floor control. For limited bandwidth, this is a way to limit number of concurrent users on the network. For example, one type of member will be not just a member but also a potential co-ordinator and repeater. Another type of members will be just normal members, the last type of member will unreliable receiver who will not ask for retransmission. If the members are categorised like that then the job of the application programmer is made a lot easier. A model like MTP/SO proposes to meet this requirement.

#### 4 LIMITATIONS OF FLOOR CONTROL

A lot of the multicast transport protocols like SRM, RMTP, MTP/SO will meet some of the requirements for floor control. Certain protocols can be customised or adopted to meet some of the requirements. However, there are some limitations of a floor control mechanism itself because of the nature of its behaviour. The principle difficulty is in achieving scalability to large group sizes. In a conference, where all members have access to the ability to request (and grant) the floor, it is necessary for all participants to know who the other participants are. Otherwise, none can see a global reason for giving someone the floor.

If the access bandwidth is small compared to network backbone bandwidth, at time  $t$ , there may be 1000 receivers in the system, however using RTCP the report

of the participants may show only first 20 participants<sup>\*</sup>. To account for congestion control a solution has been suggested in timer reconsideration for enhanced RTP scalability [Rosenberg98]. In a multimedia session which is using RTP/RTCP for transporting audio and video where RTCP rate is 1 kb/s. If all RTCP packets are 1 kb, packets should be sent at a total rate of one per second. Under steady state conditions, if there are 100 group members, each member will send a packet once every 100 seconds. However, if 100 group members all join the session at about the same time, each thinks they are initially the only group member and sends a packet at a rate of 1 per second, causing a flood of 100 packets per second or 100 kb/s, into the group.

So the effect of timer reconsideration algorithm is to reduce the initial flood of packets, which occur when a number of users simultaneously join the group. A participant P who wants to join at time  $t$  will determine the group size and it will transmit at time  $t'$ , where  $t' > t$ . So if a session has to start at 10:00 am, packets will be sent at 10:01 am, 10:02 am and so on. Therefore, at time  $t$ , the report showing the number of participants at 10:00 am will not be correct.

So the underlying technology has to support users to join a session at  $t''$  where  $t'' < t$ . In other words, if the session is programmed to be broadcast at 10:00 am, users have to join the session from 9:55 am. That requires modification of connection charges to include the traffic flow pre session.

If each participant sends messages at the rate of  $K/N$  per second, where  $K$  is the fraction of total capacity allowed for the RTCP messages, the following can be derived:

For audio, we might choose to have 1 speaker and therefore  $K$  is the capacity of that 1 flow. Typically RTCP messages might be limited to 5% of the flow, so for 20 packets per second, we would be allowed 1 message per second. Over 5 minutes, this would allow  $N$  to reach 300.

For video, we may choose to allow either one video flow or several, to save bandwidth, we probably choose the current speaker's video channel, we might be sending 100 packets per second from each and every source, which allows for  $K=5$ , or  $N$  to reach 1500 participants after 5 minutes.

## 5 OBSERVATIONS

Many protocols are proposed and implemented:

Protocols differ widely in design

Logical structure of communication pathways (ring versus tree versus none)

---

<sup>\*</sup> If the reliable protocol is distributed (e.g. in SRM/NTE) i.e. the participants can only see the local information straight away and overall statistics is an option, then this problem can be eliminated to an extent.

Group membership mechanisms and assumptions  
 Receiver-reliable versus sender reliable  
 ACK/NAK and FEC

Based on floor control requirements from a reliable IP multicast (as discussed in section 3.0) SRM will be one of the most suitable transport protocols if all the participants are multicast capable. Because SRM represents a simple and robust approach for large-scale recovery based on persistent state, suppression of duplicate NACKs and repairs, and global retransmissions. The messages specify a time-stamp used by the receivers to estimate the delay from the source, which causes global ordering. Also the model of this algorithm is distributed so that the participants list will not take too long to update. However, if the number of participants is very large, the convergence time will grow exponentially and SRM will not be the best suited algorithm.

If some of the participants in a video conference is Unicast only a tree based structure for IP multicast like RMTP or MTP/SO will be quite suitable too. In the hierarchical system, one parent node can have several Unicast only child nodes underneath it and it can Unicast the data to these child nodes. In this model the participants list can be viewed by the parent node as shown in diagram 5.

## 6 LIMITATIONS OF SRM AND MTP/SO

SRM is very efficient for retransmitting the lost packet whereas MTP is customised to take care of different classes of members in a conference. None of these protocols cater for congestion or flood of packets which will be caused by a session starting or question time for a conference for example. This sort of problem is solved RMDP or the approach taken by RTP timer reconsideration.

## 7 IDEAL PROTOCOL FOR CONFERENCE CONTROL

After discussing the pros and cons of the different protocols it seems that a reliable multicast protocol has to be able to provide:

Congestion control: Cope with sudden burst of traffic. If number of receivers are small (for example, if it is up to 100 receivers) a buffer can be provided to store the requests. Otherwise, a mechanism has to be provided where pre session traffic flow is allowed. RTP timer reconsideration is an example to deal with congestion control. Also if a user who just got the floor waits a certain amount of time before asking for the floor again will help the implosion as well.

Ordering: The point about floor control is that requestors should get a fair chance at getting the floor. The problem with the reliable multicast transport protocols is that to scale, they use techniques like SRM (random timer). What is required is a deterministic (round robin) timers for people requesting the floor at the same time. None of the transport protocols include this feature. So if a participant asked for the floor or got the floor last time, then they have to go after everyone else - i.e. that user/participant has to wait before asking for the floor again.

Reliability : Fastest way to retransmit lost/damaged packets. Not just the source, any one holding the packet will transmit the packet to the receiver require that damaged packet. SRM's retransmission strategy provides that.

Distributed control: As discussed in section 4.0, because convergence time increases as the number of users increase, there is a limit on the size of conference of known participants. A hierarchical system with just the knowledge of certain group or certain local users will be a possible solution. RMTP or STORM can provide that sort of architecture.

Simple: Multicast the status of the floor holders, a request is multicast to the group too. Any IP multicast can provide this function.

Other: May be able to cope with Unicast only receivers too. Security is provided for alternative approaches.

## 8 CONCLUSION

There are protocols like RMTP/STORM, NTE and SRM which are designed for specific applications. SRM is a robust protocol which meets a lot of the requirements for conference control. MTP and RMTP meet certain criterias too. However, these protocols need a level of customisation or a level of adaptation to be ideal protocol for conference control. This paper also looks at the limitation of these protocols and the limitation of floor control to achieve scalability. Therefore, if a reliable multicast has to be designed to meet the requirements of floor control it can be quite complicated to cater for ordering, congestion control, pre traffic flow etc. In order to keep it simple, we need a mechanism where the status of the floor holders is multicast in every few seconds to the group. If a user wish to bid for the floor, the request is multicast too. The stabilising time/converging time grows as the number of participants grow normally, so a hierarchical system will be a better solution. It is also required to provide a distributed model for retransmission and keep the status of receivers up to date.

## 6 REFERENCES

- Dommel P., Aceves JJ (1995) Floor Control for Activity coordination in Networked Multimedia Application - Proc. 2nd Asian-Pacific Conference on Communications (APCC)'95, Osaka, Japan, June 12-16, 1995.
- "FAQ - on the Mbone", <http://www.mediadesign.co.at/newmedia/more/mbone-faq.html>
- Floyd S., V. Jacobson, S. McCanne, C. G. Liu, and L. Zhang (1995) A Reliable Framework for Light-Weight Sessions and Application Level Framing - ACM SIGCOMM '95. Boston. August 30-September 1, 1995.
- Floyd S., Varadhan K., Estrin D (1998) Impact of Network dynamics on End-to-End protocols: Case studies in TCP and Reliable Multicast.- research draft "<http://www.isi.edu/~kawnan/VINT/ic98.ps>"
- ITU H.323 recommendation (1997)
- ITU draft recommendation T.120 (1997) - Data protocols for multimedia conferencing

- Levine B., Aceves-JJ (1998) A comparison of Reliable Multicast Protocols , "<http://www.ucsc.edu/b.levine>", Multimedia Systems (ACM/Springer), Vol. 6, No.5, August 1998.
- Lin, John C. and Paul, Sanjoy (1996) RMTP: A Reliable Multicast Transport Protocol, IEEE INFOCOM '96, March 1996, pp. 1414-1424.
- Ott J., Bormann C (1997) MTP/SO - Self organising Multicast Internet draft 1997, draft-bormann-mtp-so-01.txt
- Rosenberg J., Schulzrinne H.(1998)Timer Reconsideration for Enhanced RTP scalability - Internet draft -draft-ietf-avt-reconsider-00.ps
- Sudan M., R. Yavatkar, J. Griffioen (1993) A reliable Dissemination Protocol for Interactive Collaborative Applications, ACM multimedia
- Speakman T., Farincci D., Lin S.(1998) PGM specification - Internet draft draft-speakman-pgm-spec-00.txt
- Vicisano L., Rizzo L(1997) A Reliable Multicast Data distribution Protocol based on Software FEC techniques -Proceedings of the 4<sup>th</sup> IEEE workshop on the architecture and Implementation of High Performance Communication systems (HPCS 97)
- Vicisano L., Crowcroft J., Rizzo L.(1998) TCP Like congestion control for layered multicast data transfer - Proceeding of INFOCOM 1998
- X.Rex Xu, Zhang H, Yavatkar R (1997)Resilient Multicast Support for Continuous media applications "<http://research.ivv.nasa.gov/RMP/links.html>", in Proc. International Workshop on Network and Operating System Support for Digital Audio and Video(NOSSDAV), St. Louis, May 1997
- Zheng, W, Crowcroft J, Diot C. and Ghosh A (1997) Framework For Reliable Multicast Application Design, HIPPARCH 97.

## AUTOBIOGRAPHY

Nadia Kausar is a PhD student in Computer Science Department in University College London. Nadia is looking at various multiuser, multimedia system architectures over various networks and IP telephony for last 2 years. The work has initially focused on a comparison of the ITU's conferencing designed to run on circuit-switched networks and the IETF's MMUSIC conference control architectures with a view to extracting a generic set of modules to run over the Internet. Nadia graduated with a first class degree in Computer Systems in 1994 from University of Westminster. Nadia's main supervisor is Prof. Jon Crowcroft. This work is also supervised by Ian Marshall in BT Labs, Ipswich.

Jon Crowcroft is a professor of networked systems in the Department of Computer Science, University College London, where he is responsible for a number of European and US funded research projects in Multi-media Communications. He has been working in these areas for over 18 years. He graduated in Physics from Trinity College, Cambridge University in 1979, and gained his MSc in Computing in 1981, and PhD in 1993. He is a member of the ACM, the British Computer Society and a Fellow of the IEE and a senior member of the IEEE. He is a member of the IAB and general chair for the ACM SIGCOMM. He is also on the editorial team for the ACM/IEEE Transactions on Networks. With Mark Handley, is the co-author of WWW:Beneath the Surf (UCL Press); he also authored Open Distributed Systems (UCL Press/Artech House).

# An Architecture for Conference-Support using Secured Multicast

*Thomas Hardjono, Naganand Doraswamy and Brad Cain*  
*Bay Architecture Laboratory, Bay Networks*  
*3 Federal Street, Billerica, MA 01821, USA*  
*Tel: +1-978-916-4538, Fax: +1-978-916-0620,*  
*{thardjono,naganand,bcain}@baynetworks.com*

## Abstract

The current work argues that from a security perspective there is much to be gained by employing a “secured” IP multicast at the Network layer to support the formation and management of secure conferences at the Application layer. A secured IP multicast -- with group authentication and confidentiality -- already achieves a reasonable level of security, and therefore fulfils a large part of the basic requirements of secure conferencing. If host-to-host authentication and confidentiality has been achieved through an N-to-N multicast that has been secured, then to a large extent the basic security needs of conferencing has been satisfied. What remains would be for the other conference-specific security requirements to be satisfied using methods which are particular to a given conference scheme, such as cheater detection/identification methods based on cryptographic techniques. In the current work we propose an architecture called the Multicast/Conference Security Architecture (MCSA) to facilitate the use of (a secured) IP multicast at the Network layer for establishing (a secured) conference at the Application layer.

## Keywords

Secure multicast, secure conferences, cryptography, routing protocols, key management

## 1 INTRODUCTION

The issue of group-oriented security has been a topic of interest in the field of computer and network security for the last two decades. This has been facilitated

by a number of factors, including the growth of the Internet, the development of cryptosystems (in particular, public key cryptosystems), and the development of hardware and software for desktop level computing. Increasingly users are using the Internet not only for exchanging messages, but also for more complex interactions and as a forum for decision-making.

In this paper we attempt to bring together the main areas of research and development related to group-oriented security. The first area consists of solutions and schemes which are known collectively under *group-oriented cryptography*. These cover, among others, conference key distribution systems (eg. Ingemarsson, Tang & Wong 1982, Koyama & Ohta 1987, Steiner, Tsudik & Waidner 1996), digital multisignature schemes and secret sharing schemes (Simmons 1992). Most of the conference key distribution schemes that have been proposed require extensive cryptographic operations and have been designed with the Application layer in mind. These employ user authentication techniques either separately or integrated into the conference key distribution scheme. Some rely on the use of smartcards (eg. Koyama et al. 1987) as a user authentication technique and as a medium to store the conference security parameters.

The second area of development closely related to group-oriented communications is multicast, more specifically IP multicast. Here, group security is to be achieved through the distribution and management of cryptographic keys at the Network layer, using approaches which we broadly term *group key management* (GKM) protocols (Mitra 1997, Harney & Muckenhirn 1997, Ballardie 1996, Harkins & Doraswamy 1997).

In this paper we argue that from a security perspective there is much to be gained by employing a "secured" IP multicast at the Network layer to support the formation and management of secure conferences at the Application layer. A secured IP multicast -- with group authentication and confidentiality -- already achieves a reasonable level of security, and therefore fulfils a large part of the basic requirements of secure conferencing (see Section 2.1). If host-to-host authentication and confidentiality has been achieved through an N-to-N multicast that has been secured, then to a large extent the basic security needs of conferencing has been satisfied. What remains would be for the other conference-specific security requirements to be satisfied using methods which are particular to a given conference scheme, such as cheater detection/identification methods based on cryptographic techniques. Other benefits that would emerge include the increased efficiency of the conference formation and the possibility of the simplification of the conference key distribution schemes being employed.

In the current work we propose an architecture to facilitate the use of (a secured) IP multicast at the Network layer for establishing (a secured) conference at the Application layer. The purpose of the architecture is to introduce components at the Session layer that will coordinate the creation of multicast sessions, the

obtaining of group-keys for the groups, and the “mapping” of the conference instance to the multicast instance, and other tasks. The architecture must also allow different secure conference key distribution schemes to be used, independent of the multicast protocol and group key management protocol at the Network layer.

In the remainder of the paper, we define a “conference” as the N-to-N communications occurring at the Application layer (or originated from events at the Application layer). Similarly, we define “multicast” or “IP multicast” as the N-to-N communications occurring at the Network layer. We use the term multicast “session” to include all multicast “groups” related to a session. Thus, for example, a session may have an audio group and a video group (as in the Mbone), and both would be considered together when dealing with security. Consistent with this convention, we thus distinguish between a “conference key” for a conference and a “group key” for a multicast instance. For simplicity of the discussions, in either of the two cases we assume that the key is a private key (ie. symmetric cryptosystems) and we assume that the key is used to encipher traffic among the parties involved. Other variations on the type of key can certainly be employed depending on the needs of the circumstances.

In Section 2 some background is provided, looking from the perspectives of both the Application layer and the Network layer. This is followed by the proposed architecture in Section 3. Section 4 closes the paper with some remarks and conclusions.

## 2 PERSPECTIVES ON CONFERENCES AND MULTICAST

Research and developments in the area of group-oriented security in the past two decades has largely focused on two major directions which can be viewed from the *Application-layer perspective* and from the *Network-layer perspective*, reflecting the two main locations in the communications software architecture where solutions have been suggested.

### 2.1 The Application Layer Perspective

Much of the research efforts carried-out on the Application layer revolved around the use of cryptographic techniques to achieve a fair and secure method to distribute cryptographic keys to participants of a conference. The key belonging to the conference is then to be used to secure (eg. encrypt/decrypt and/or authenticate) the messages exchanged between the conference participants. To these schemes we apply the broad term *Conference Key Generation and Distribution System* (CKGDS), which may or may not incorporate user-authentication. The conference membership is usually determined by the piece of secret information carried by (or assigned to) the users. Joining a conference can be signified by the user indirectly



applying the portion of his/her secret information towards the conference-key computation. (For example, the user may apply his/her secret key to a circulating irreversible “token” that accumulates the keys). CKGDSs are often coordinated by a *Conference Coordinator* (eg. the participant that initiated the conference or a trusted third party). Other approaches employ a trusted third party to fully generate and deliver the conference key.

There are a number of security requirements for secure conferences. Some requirements that are of interest to the current work include:

- *Source identity and source authentication*: a member of the conference must be able to verify the source (identity of the sender) and authenticate the message from the sender. This assumes that user authentication has been enforced.
- *Data confidentiality*: encryption of data for confidentiality must be available for all traffic should the conference members decide to use such means.
- *Participation non-repudiation*: depending on the nature of the conference, members of the conference must not be able to repudiate their presence in a particular conference. This is crucial when the conference arrives at a decision which is binding to all members who are present, and who should not, therefore, be able to deny this fact at some future time.
- *Sender/receiver non-repudiation*: both senders and receivers in a conference must not be able to repudiate the fact that they sent or received messages respectively.
- *Cheater detection and identification*: any members (or intruders) that cheat must be able to be detected and identified. There is a range of behaviours that can be considered as cheating. A typical example would be a member that wishes to participate in decision-making a conference, but who would like to avoid the commitments resolved in that conference, by way of not submitting the correct information (eg. security parameters) at the creation of the conference. Other examples include masquerading as other members of the conference.
- *Joining conferences securely*: new parties that are permitted to join a conference (eg. by policy) should be able to do so in a manner that does not compromise or reduce the security of the existing conference participants.
- *Secure ejection of a member*: when a conference wishes to eject a member, a secure method must be used, both to arrive at the consensus with regards to the ejection decision and to actually carry-out the ejection. The ejection of a member should not in any way effect the security of the remaining members.

Although outside the scope of the current work, another issue that is commonly related to conferencing is the *anonymity* of the participants of the conference. Should it be required, anonymity can be achieved at the conference level using the appropriate secure conferencing scheme where the identity of the actual user is hidden using a *pseudonym* (Chaum 1981), and where the identity and the pseudonym is linked through other pre-conference means (such as smartcards, for

example, in the scheme of Koyama et al. (1987)). Although in most circumstances the end-users of the conference service are humans who require the identity of the peers to be known, there are circumstances where pseudonyms can be used and be made legally binding should the conference require it.

## 2.2 The Network Layer Perspective

In contrast to the efforts based on extensive cryptographic operations to satisfy the security requirements of conferencing, the developments that strive to find a solution at the Network layer have originated from the need and desire to make IP-multicast itself secure, independent of the applications that may employ it. These have taken the form of *group key management* (GKM) protocols, and have largely focused on providing practical and implementable solutions based on the realization that the network has true physical limitations and that the user-response quality represents a major factor in the design. The GKM approach is driven by the realization that although some security mechanisms have been introduced into applications that make use of IP-multicast, in itself IP-multicast does not have any authentication and/or confidentiality features.

A number of GKM protocols have been proposed (eg. Mittra (1997), Harney et al. (1997), Ballardie (1996), Harkins et al. (1997)). Here, the idea is that the GKM protocol (running at each relevant host) distributes keys to all hosts of a multicast session. Each group is assigned a unique key, and all traffic within the group would then be encrypted using the group-key. Only group-authentication is thus afforded, not sender-authentication.

One of the fundamental aims of these GKM protocols -- which is an aim that we also subscribe to -- is to separate the group key management for multicast from the multicast service itself, thereby providing independence of the GKM from the multicast protocol that happens to be in use (eg. CBT of Ballardie, Francis & Crowcroft (1993), MOSPF of Moy (1994), and others). Although there have been a number of proposals for a group key management protocol, none at the moment are being used on a wide scale on the Internet.

Other developments towards securing communications at the Network layer have emerged in the form of IPSEC (Atkinson 1995) and its related security technologies such as the Internet Security Association and Key Management Protocol (ISAKMP) (Maughan & Schertler 1997) and the Internet Key Exchange (IKE) (Harkins & Carrel 1998). IPSEC and IKE are unicast technologies that aim to secure communications between a sender and a receiver at the Network layer. This is achieved through the creation of *security associations* (SA) -- linked to a *Security Parameters Index* (SPI) -- which identifies all the parameters (ie. encryption algorithm, algorithm mode, encryption keys, etc) required for the two parties to securely communicate. However, IPSEC is designed for unicast between

one sender and one receiver, and therefore is not fully satisfactory for the security needs of multicast. In particular, the current version of IPSEC does not provide for the creation of a single security association for an entire multicast group.

## 2.3 Classification of Multicasts and Conferences

In this section we broadly classify groups from the perspective of the Network layer and the Application layer, realizing in full that the various attributes of groups have different interpretations in different circumstances. It follows that there is no one solution that can solve the security needs of both multicast and conference. Hence our approach of an architecture which can support the various parameters of multicast and conferences.

### 2.3.1 Network Layer Perspective

#### Receiver view:

##### *Open multicast*

- Any host is free to join a group and to receive the group's traffic.
- Joining does not require explicit permission.
- The receiver host requests its local subnet router to join a given group (eg. using IGMP or similar group-membership protocol).

##### *Closed multicast*

- Explicit permission must be requested to join a group (eg. to the initiator of the group).
- Mechanisms must be employed to enforce permissions (for example, encryption of traffic within the group).

#### Sender view:

##### *Open multicast*

- Any host can initiate/create a new group<sup>1</sup>.
- Any host can send to a group without being a member of the group<sup>2</sup>.
- Any host can send to any group.

---

<sup>1</sup> Although a host can create a group, technically speaking the host does not have to be a sender or receiver in the group. For simplicity, we assume that a host that creates a group is at least a sender.

<sup>2</sup> This may or may not be desirable, as it may allow denial-of-service attacks to the group. However, the notion of an authorized non-member sending to a group may have its uses and benefits.

*Closed multicast*

- Not everyone can create a group, and it is desirable to have mechanisms to limit who can create/initiate new groups.
- Only members of a group can send to the group.
- Traffic may be enciphered as a way to enforce explicit permissions.

*2.3.2 Application Layer Perspective***Receiver (Sender) view:***Open conference*

- Any user can initiate a conference.
- Any user can join or receive (send) provided he or she reveals his/her verifiable identity to the conference Coordinator and other conference participants.
- Presence at the conference does not bind a participant to the outcomes of the conference.
- The user need only submit parameters that are sufficient for identification purposes.

*Closed conference*

- Any user can initiate a conference.
- By definition, only limited individuals can receive from (send to) the conference.
- An invitee must make explicit request to the conference Coordinator.
- The identity of an invitee must be verified before he or she can receive (send).
- The invitee must submit cryptographic parameters as part of the conference key generation and distribution scheme.
- Participation in the conference must be provable through the key generation and distribution scheme.
- A participant must not be able to repudiate his/her presence and participation in the conference.
- The conference outcome is legally binding.

**3 MULTICAST SUPPORT FOR CONFERENCING**

In the current work we propose an architecture to facilitate the use of (a secured) IP multicast at the Network layer for establishing (a secured) conference at the Application layer. The purpose of the architecture is to introduce components at the Session layer that will coordinate, among others, the creation of multicast

groups, the obtaining of group-keys for the groups, and the “mapping” of the conference instance to the multicast instance. The aim is also to provide protocol independence, in the sense that different secure conferencing schemes at the Application layer can be used independent of the multicast protocol and group key management protocol at the Network layer.

One important assumption in the proposed architecture is that the underlying multicast service model (and its associated routing structures) allows a host who is part of a group to also transmit messages to the group. This can be achieved by using a multicast protocol that allows a receiver-host to also transmit messages to the group. Thus, in effect the underlying multicast protocol performs the N-to-N multicast. Most IP multicast protocols today allow this to occur.

In the case that the multicast protocol is limited to providing unidirectional transmission, then an *overlay* of N instances of these 1-to-N multicasts must be created and managed. That is, N separate multicast “trees” (emanating from a source to the receivers) must be created. This overlay will result in the creation of a total of  $N^2$  security associations. Due to its large resource consumption, we will not consider this second approach any further.

### 3.1 Multicast Conferencing Security Architecture

The current work approaches the issue of providing support for conferences using (a secured) multicast by introducing the *Multicast/Conferencing Security Architecture* (MCSA) that provide selectable components which reside at the Session layer. The MCSA components together act as an intermediary between the conference application program at the Application layer and the multicast-related protocols (host-side or client-side) -- such as the Group Management Protocol (GMP) and the Group-Key Management (GKM) protocol -- at the Network layer (Figure 1). Here we use the general term GMP to denote the group membership and management protocol that is being employed (for example, IGMP (Deering 1989)). The IPSEC and IKE protocols (Atkinson 1995, Harkins & Carrel 1998) are also used -- directly or through the GKM protocol -- for host-to-host key exchange and data confidentiality.

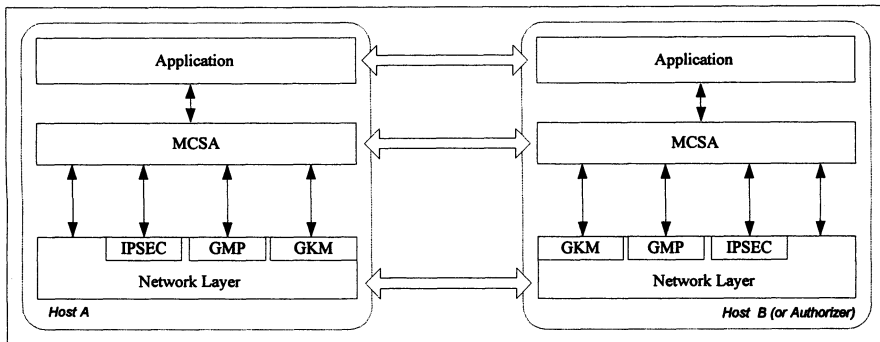


Figure 1: The Multicast/Conference Security Architecture

In practice the GKM protocol typically has a central point which authorizes and distributes keys for the group. Here we will broadly refer to that entity as the *Authorizer*, which may generally correspond to the *Group Key Controller* in the work of Harney et al. (1997), to the *primary core* in the CBT-based solution of Ballardie (1996), or to the *Key Distributor* (KD) in the scheme of Harkins et al. (1997). The Authorizer entity can be a router, a server or other devices.

In order to participate in the MCSA, the Authorizer entity must contain the components of the MCSA so that peer-level interaction would be possible. The Authorizer is assumed to run the GKM protocol to establish group-keys and perform high-level tasks, including certificate management, user access control, policy implementation, and others. The notion of an Authorizer is beneficial also from the point of view of security and access control policies, since the Authorizer can embody these policies. The Authorizer within a subnet can in fact be a subset of the *policy hierarchy* governing the entire autonomous system.

From the Network layer's perspective the Authorizer generates group-keys which are used by the group members to encrypt the payload within the multicast packets of a given group. Each group is assumed to have a secret (symmetric) key which is obtained by each member-host through a secure association (SA) with the Authorizer. Thus, further implied is the fact that each member-host must have a *distinguished name* (DN) (Adams and Farrell 1998) and must have a certificate before any security association can be established (Atkinson 1995). Hence, in our architecture we assume that each host has already been assigned a distinguished name and a certificate by the certification authority (CA).

### 3.3 Components of the Architecture

The MCSA consists of a number of components (Figure 2) which orchestrate the interaction between the multicast and the conference events. All parties involved in

the conference and multicast is assumed to employ the MCSA. Furthermore, we assume that the Authorizer contains all the peer elements at the Application layer and at the Network layer. The arrows in the diagram are simplified to denote interaction, both in terms of control/invoke and data/control flow.

The Conference Session Manager (CSM) and the Multicast Session Manager (MSM) are the two components that look after the relevant events occurring at the Application layer level and at the Network layer respectively (Figure 2). The CSM works in conjunction with the conference application and the conference key generation and distribution system (CKGDS). The CSM maintains the state information for each conference of which the user is a participant, and it maintains a database of the corresponding conference keys. Although the CSM does not actually use the conference keys, it has access to the database in order to maintain a correspondence between the instance of the conference (since the user maybe involved in several simultaneously) and the key for that conference.

The Multicast Session Manager (MSM) cooperates with the Conference Session Manager (CSM) in maintaining a correspondence between a conference instance and its underlying multicast instance. Depending on whether a host is the initiator of the multicast group or a receiver/sender in the group, the MSM has a number of tasks related to the creation of the multicast and the securing of it. The MSM is responsible for the initiation of the multicast group as a response to requests from the conference application. Through the session directory (SD) it coordinates the announcement of the new multicast group. It communicates with the peer MSM at the Authorizer in order to notify the Authorizer of the new group and request a new group key.

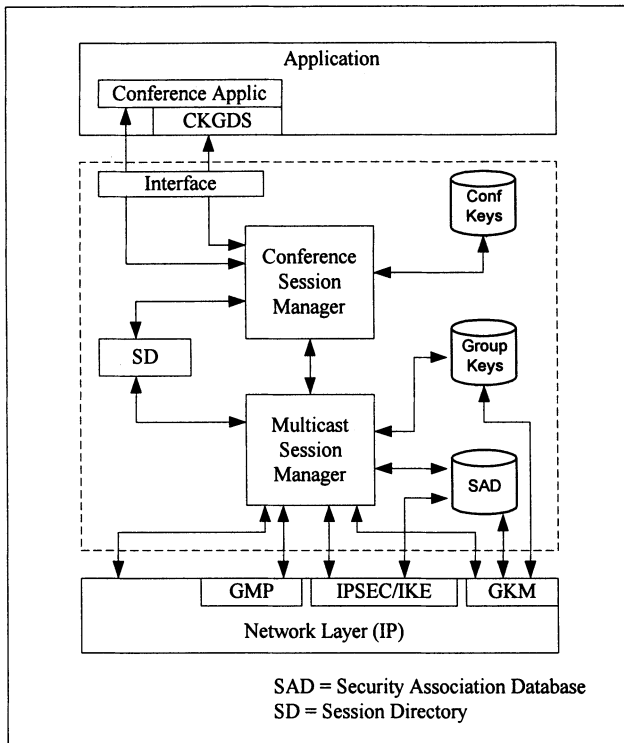


Figure 2: Components of the MCSA

A user that wishes to join a conference must instruct its host (implementing the MCSA) to first join the multicast group corresponding to that conference. Joining the multicast group can be achieved through the GMP. After the host becomes a group member, it must open a secure association with the Authorizer in order to obtain the corresponding group key. The MSM must maintain a mapping between the multicast instances (of which the host is a member) and the security association contained in the Security Association Database (SAD), the SAD being part of the IPSEC definition. Through the security association, a copy of the group key is obtained from the Authorizer and it is stored in the group key database. This group key is used to encipher (decipher) traffic at the Network layer destined for (received from) the multicast group.

Although not clearly shown, ideally the conference application and the corresponding CKGDS should deal with the MCSA through a suitable interface or API. Such an API should be usable with a variety of CKGDSs and may even be extendable to other applications that exhibit conference-like behaviours and security requirements.



### 3.4 Initiator and Sender/Receiver Interactions

In the current architecture, we assume that at the conference level, the user/application initiating the conference is the Coordinator of the conference and the contact-point for other users wishing to join or leave the conference. The issue of the ejection of conference members is also the responsibility of the Coordinator.

In the following we consider the MCSA as a unit from the perspective of a sender/receiver and an initiator of a conference. We assume that the initiator host is also where the conference Coordinator resides.

#### Initiator MCSA:

- Upon a conference-formation request from the application/user, the MCSA invokes the session directory (SD/SDP) to announce a new multicast session. The announcement must contain the identity of the Authorizer from which other hosts can obtain the group key. The announcement must also indicate that the multicast session will be part of a conference soon to be created. The application/user may also provide a list of acceptable (non-acceptable) users for the conference and the associated access control information.
- The MCSA creates a security association with the Authorizer and notifies the Authorizer about the new multicast session. The MCSA also provides it with access control information (for access at the user level and the host level). The Authorizer prepares a group key for the multicast session.
- The MCSA uses the security association to obtain a copy of the group key.
- The MCSA invokes IPSEC to encipher all subsequent traffic destined for the multicast session.
- After a given period of time (to allow other hosts to become members of the multicast session) the MCSA initiates the transmission of a conference-call message through the multicast session, with the announcement details being encrypted using the group-key. It names its own CKGDS as the Coordinator of the conference.
- The MCSA waits for the conference-call-responses from the members of the multicast session, and notifies its conference application and CKGDS about the group-members that request to join the conference.
- The MCSA notifies the CKGDS that all peer MCSA are waiting for the conference key generation and distribution to commence. The Coordinator CKGDS (at the Initiator host) then begins the conference key generation and key distribution phase, culminating in each application having a copy of the

conference key. Note that all traffic during (and after) the conference key generation/distribution are encrypted at the Network layer using the multicast group key.

### **Receiver/Sender MCSA:**

- Upon user request the MCSA invokes the session directory tool to determine active multicast sessions. (ie. start time, duration, IP addresses, formats, etc). Included here is the address of the Authorizer. As an aside, the user may also request the Authorizer to provide it with a copy of the Authorizer's certificate (signed by an acceptable global authority) to prevent masquerading.
- The MCSA notifies the user/application about the active multicast sessions and asks the user to select the multicast sessions to join.
- The MCSA invokes the GMP to notify the local subnet router about the request to join a particular multicast session(s) which comprise a conference. (This step depends to a large extent on the multicast protocol being employed).
- Upon its host becoming a member of the multicast session, the MCSA invokes the GKM to obtain the group key. One way to obtain the key is to create a security association (SA) with the Authorizer, and then to use the resulting secured channel to download a copy of the key.
- The MCSA invokes IPSEC to decipher (encipher) all subsequent traffic received from (destined for) the multicast session using the group key.
- Upon seeing a conference-call message (issued through the peer MCSA at the Coordinator host) the MCSA responds to the call and notifies its own CKGDS of the ready-status of the Coordinator CKGDS. After some conference parameter negotiations, the CKGDS participates in the conference key generation and distribution scheme.

Note that in the current architecture the conference application can still employ its own mechanisms for confidentiality and authentication at the Application layer. This approach may be preferable in certain circumstances, where the security requirements are more stringent and where user-to-user security must be established (eg. using smartcards).

In the current architecture, the group key at the host level only affords *group authentication*. That is, other member-hosts can be assured implicitly only that the data originated from a valid group-member (unless other additional means is employed). If *sender authentication* at the Network layer is also required in order

to identify the source-host, then each member-host must embed its signature as part of the payload. This is because IPSEC does not provide explicit means to include signatures for authentication for each data packet. Digital signatures can be employed, while less explicit signing mechanism are also available.

## 4 REMARKS AND CONCLUSION

In this paper we have argued that from a security perspective there is much to be gained by employing a “secured” IP multicast at the Network layer to support the formation and management of secure conferences at the Application layer. A secured IP multicast -- with group authentication and confidentiality -- already achieves a reasonable level of security, and therefore fulfils a large part of the basic requirements of secure conferencing.

The current work has viewed and discussed group-oriented security from two perspectives, namely secure conferencing at the Application layer (via conference key generation and distribution systems) and group key management protocols at the Network layer. It has also attempted to classify multicasts and conferences in order to find similarities and differences in behaviour at the two layers.

Following from this the Multicast/Conferencing Security Architecture (MCSA) was proposed that provided a way to identify components at the Session layer that could act collectively as an intermediary between the conference application program at the Application layer and the multicast-related protocols (host-side or client-side) at the Network layer. Part of the internal task of the MCSA is to map instances of conferences to that of multicasts in a secure fashion with the aid of an Authorizer entity. The interaction of the relevant parties in the multicast and conference has also been briefly outlined.

There are still a number of open problems in the area of group-oriented security, particularly with respect to practical GKM protocols. These include the introduction of a *group security association* (GSA) for IP multicast in the spirit of IPSEC, the design of a GKM protocol for inter-domain key distribution suitable for the various IP-multicast applications, and others. These, as well as other issues specific to the MCSA, will be the directions for future work.

## 5 ACKNOWLEDGMENTS

We thank Jim Luciani for the important input and support for the current work. We also thank the anonymous referees for their extensive comments and insights into the issues discussed in the paper.

## REFERENCES

- Adams, C., and Farrell, S. (1998) Internet X.509 public key infrastructure certificate management protocols, March 1998. draft-ietf-pkix-ipki3cmp-07.txt available at <http://www.ietf.org>.
- Atkinson, R. (1995) Security architecture for the internet protocol. RFC 1825, IETF, August 1995.
- Ballardie, T. (1996) Scalable multicast key distribution. RFC 1949, IETF, 1996.
- Ballardie, T., Francis, P., and Crowcroft, J. (1993) Core based trees: An architecture for scalable inter-domain multicast routing. In Proceedings of ACM SIGCOMM'93 (San Francisco, 1993), ACM.
- Chaum, D. (1981) Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM* 24, 2 (1981), 84--88.
- Deering, S. (1989) Host extensions for IP multicasting. RFC 1112, IETF, 1989.
- Harkins, D., and Carrel, D. (1998) The internet key exchange (IKE), March 1998. draft-ietf-ipsec-isakmp-oakley-07.txt available at <http://www.ietf.org>.
- Harkins, D., and Doraswamy, N. (1997) A secure scalable multicast key management protocol, November 1997. draft-ietf-ipsecsecond-00.txt.
- Harney, H., and Muckenhirn, C. (1997) Group key management protocol (GKMP) specification. RFC 2093, IETF, July 1997.
- Ingemarsson, I., Tang, D. T., and Wong, C. K. (1982) A conference key distribution system. *IEEE Transactions on Information Theory* IT-28, 5 (1982), 714--720.
- Koyama, K., and Ohta, K. (1987) Identity-based conference key distribution systems. In *Advances in Cryptology - CRYPTO'87* (Lecture Notes in Computer Science No. 293) (1987), Springer-Verlag, pp. 175--184.
- Maughan, D., and Schertler, M. (1997) Internet security association and key management protocol (ISAKMP), July 1997. draft-ietf-ipsec-isakmp-08.txt available at <http://www.ietf.org>.
- Mitra, S. (1997) The Iolus framework for scalable secure multicasting. In Proceedings of ACM SIGCOMM'97 (1997), ACM, pp. 277--288.
- Moy, J. (1994) Multicast extensions to OSPF. RFC 1584, IETF, 1994.
- Simmons, G. J. (1992) An introduction to shared secret and/or shared control schemes and their application. In *Contemporary Cryptology: The Science of Information Integrity*, G. J. Simmons, Ed. IEEE Press, 1992, pp. 441--497.
- Steiner, M., Tsudik, G., and Waidner, M. (1996) Diffie-Hellman key distribution extended to group communications. In Proceedings of the 3rd ACM Conference on Computer and Communications Security (New Delhi, March 1996), ACM.

# SELDOM: A Simple and Efficient Low-cost, Delay-bounded Online Multicasting

*Tawfig Alrabiah and Taieb F. Znati*  
*Department of Computer Science*  
*University of Pittsburgh*  
*Pittsburgh, PA 15260*  
*{tawfig, znati}@cs.pitt.edu*

## Abstract

With the advent of multimedia applications, the support of *on-line* multicasting with quality of service (QoS) guarantees has gained considerable attention in the field of communication networks and distributed systems. The objective of this paper is to investigate on-line QoS-based routing and path establishment schemes to support point-to-multipoint connections in wide area networks. We propose SELDOM, a Simple and Efficient Low-cost Delay-bounded Online Multicasting scheme to support on-line multicasting. The scheme is particularly tailored to networks in which group membership changes frequently.

The approach taken by the scheme is unique in the sense that, given a set of QoS requirement specifications of each multicast node and the current status of the network links, SELDOM finds a minimum cost multicast tree that meets these QoS specifications of the supported group members. The scheme handles *join* requests dynamically by determining the least cost path which satisfies the required delay bounds to which the new group member is to be attached. On the other hand, to handle a *leave* request, the scheme seeks to limit the rearrangement required in order to reduce the disturbance such a request may cause to current members of the group. The worst time complexity of SELDOM is  $O(n^2)$ .

## Keywords

online multicast routing, steiner tree, quality of service

## 1 INTRODUCTION

Recent advances in high-speed networking technology have created opportunities for the development of a wide spectrum of sophisticated multimedia applications which generate, integrate, process, store, and distribute time dependent and time independent media. Typical applications include video conferencing, computer supported collaborative work, and limited video broadcasting. These applications are characterized by a wide spectrum QoS requirements and the need for *group communications* among multiple end-points.

Support of QoS-based group communication in multimedia environments requires the development of efficient and cost-effective *multicast* algorithms. The ability to perform such a task is becoming a major requirement for computer networks that support multimedia applications. To increase the fraction of accepted multicast sessions, the network must use the minimum amount of network resources while guaranteeing the sessions' QoS requirements. From the routing point of view, an efficient multicast algorithm must only replicate packets when necessary, namely at the branching points at the tree.

In the past, the bandwidth required by applications was small and the applications' QoS requirements were not as stringent as those of current multimedia applications. Hence, simple multicast algorithms were used to manage the network resources. However, with the advent of multimedia applications, developing efficient multicast algorithms is becoming increasingly important. To increase the rate of accepted multicast sessions, new algorithms that minimize the amount of replicated traffic exchanged during multimedia multicast session must be developed. These algorithms must guarantee the stringent QoS requirements of multicast sessions while minimizing the cost of the multicast trees used to exchange the resulting traffic.

The multicast group set can be known before setting up the multicast routing tree. In this case, the problem is called the *off-line* multicasting problem. What is required in this case is an algorithm that, given a set of QoS requirement specifications, current status of the network links, and the multicast set, find a *Minimum Cost Multicast Tree* (MCMT) that meets the QoS specification of the multicast tree nodes.

Multicast applications such as teleconferencing, distance learning, collaborative work, and data distribution may require the ability to support dynamic sessions. Dynamic sessions are characterized by their members' ability to join or leave in a dynamic fashion. The sudden and unexpected arrival and departure of session members makes the multicast problem an on-line problem where routing decisions have to be made on-line while the multicast session is in progress. Hence, the multicast group set is not known prior to setting up the multicast session. This problem is called the *Online Minimum Cost Multicast Tree* (OMCMT) problem.

The objective of the multicast problem in a multimedia network is to build a low cost tree that bounds the source-destination delay. That is, given a

graph  $G = (V, E)$ , where each link is assigned cost and delay, a source node  $s$ , a multicast set  $D$ , find the lowest cost multicast tree that bounds the source-destination delay. This problem is NP-complete. That can be easily proved by setting up the delay bound to infinitely which reduces the problem to the Steiner tree problem. The Steiner tree problem is a well known NP-Complete problem [19]. Several exact solutions to the Steiner tree problem have been proposed in [8, 19]. All proposed exact solutions, however, require exponential execution time. This prompted the development of several polynomial heuristics for approximate solutions. A survey of these heuristics, as well as exact algorithms, for Steiner problems in networks is provided in [19].

The rest of the paper is organized as follows: we start with reviewing some of the approaches and algorithms proposed to provide an approximate solution to the off-line, low-cost, delay-bounded multicast trees. After that, a discussion of low-cost online multicasting algorithms, which update and maintain multicast trees dynamically in response to join or leave requests, is presented. Section 3 introduces SELDOM which is a new proposed algorithm for the online, low-cost, delay-bounded multicast problem. A conclusion of this work will be presented in the last section.

## 2 RELATED WORK

Based on their design objectives, the multicast algorithms proposed in the literature can be viewed as members of one of three possible classes. The first class includes algorithms which are designed to accommodate an Internet based environment. The second class includes off-line algorithms which aim at reducing the cost of the multicast tree, while bounding the end-to-end delay. The third class includes algorithms which deal with on-line multicasting. These algorithms are reviewed next.

### 2.1 IP-based Multicast Protocols

The Internet community proposed different algorithms to create multicast trees, including Distance Vector Multicast Routing Protocol (DVMRP), Multicast Open Shortest Path First (MOSPF), Protocol Independent Multicasting (PIM), and Core Based Trees (CBT) [13, 14, 7, 3]. DVMRP is built on top of RIP (Routing Information Protocol), a distance vector protocol, which is not efficient in detecting loops and link failures quickly. MOSPF uses Open Shortest Path First (OSPF) to maintain a current image of the network topology. CBT builds a single distribution tree, formed around a focal router which is called the core. The major drawback of CBT is the concentration of traffic at the core of the tree. Hence, CBT is vulnerable to core failure which can partition the tree. The Protocol-Independent Multicast (PIM) addresses both dense (PIM-DM) and sparse (PIM-SM) environments [7]. PIM-DM is en-

visioned to be used in an area where group membership is dense. PIM-DM is similar to DVMRP except the unicast routing information is imported from the existing unicast protocol rather than incorporating it in the implementation of the unicast protocol. PIM-SM creates a center in the tree which is called a Rendezvous Point (RP). Each multicast group has a default router as its RP. New receivers join the tree through the RP. A receiver can switch from the shared tree to a source based tree. Upon switching the source prunes itself from the shared tree.

The above Internet multicast algorithms are designed to work specifically with the current IP environment and to take advantage of the IP routing protocols such as RIP and OSPF. The design objectives of these algorithms focused on issues related to scalability and reduced communication overhead, but did not address the QoS requirements of multimedia applications.

## **2.2 Off-line, Low-cost, Delay-Bounded Multicast Tree Problem**

In addition to low cost, multimedia applications have different demands in terms of bandwidth, reliability, delay and jitter. A key property of multimedia data is its time dependency. The support of sustained streams of multimedia objects, over a period of time, requires the establishment of reliable, low delay and low cost source-to-destinations routes. Nevertheless, the objective is not to develop a strategy which produces the lowest possible end-to-end delay, but a strategy to ensure that the data traffic arrives within its delay bound, thereby allowing a tradeoff between delay and cost. Thus, the objective is to produce a tree that has minimal cost among all possible trees that bound end-to-end traffic delay between all source-destination pairs.

Many heuristics were developed for the low-cost unbounded-delay multicast problem [20, 2]. However, there are few attempts to develop low-cost bounded-delay multicast heuristics. In the following, we review off-line, low-cost, delay-bounded, multicast heuristics. A simple approach to solving this problem is to use a tree that is composed of the least delay paths from the source to the multicast nodes. Such an approach will always find a solution that conforms to the delay bounds if one exists. This approach, however, does not take into consideration any cost optimization. A different heuristic for solving the delay constrained multicast tree is to use the constrained shortest path tree [15]. This heuristic first builds a tree composed of the shortest cost paths to the destinations. If the end-to-end delay to any group member is violated, the least delay path will be used instead.

A dynamic programming approach was suggested by Kompella, Pasquale, and Polyzos [12]. This heuristic assumes that link delays are represented by integer values. The heuristic begins by finding the least cost bounded path from each multicast node to another. Then, it uses the minimum spanning



tree algorithm to connect the multicast nodes without violating the end-to-end delays. The complexity of this approach depends on the granularity of the delay values. If the granularity of the delay is very small then the complexity will be large.

An iterative optimization approach to the minimum delay tree was suggested in [21]. The algorithm starts with the minimum delay tree. Then, it replaces the relay paths with lower cost paths without violating the delay bound. It continues until the cost of the tree cannot be reduced further.

A simple heuristic based on the Simple Path Heuristic (SPH) [16] was proposed in [1]. The heuristic *Least Cost First* (LCF) decouples the cost optimization from bounding the delay by building a low cost tree incrementally and then checking the delay bound requirements. The node with the least cost path to the tree is selected. If the path to that node violates the delay bound, the least-cost delay-bounded path out of the possible low cost paths from the tree to that node is used instead. This process continues iteratively until all multicast nodes are included. Failure to include all multicast nodes implies that no multicast tree which satisfies the QoS requirements for all multicast nodes exists. The analysis shows that the performance and complexity of LCF heuristic are comparable to those of the SPH approximation if the number of delay violations remains moderately small.

Most of the above algorithms were designed for undirected networks. However, they can be implemented in a directed network. Also, they are only designed for static multicast trees (off-line). In the rest of this paper we discuss online multicast tree. All of the online heuristics that we are aware of address low cost online multicasting with no consideration to delay bound. In the following we will discuss some of these online multicasting heuristics.

### 2.3 Online Heuristic Algorithms

The multicast group membership in typical multimedia settings, such as on-line video conferencing or multimedia group authoring, dynamically changes as new members request to join the group or current members request to leave the group. Therefore, supporting dynamic multicast applications efficiently requires adding or deleting members to the multicast tree efficiently and transparently to the other multicast members. While many research works have addressed static multicast group communications in WANs, very little research has considered the dynamic version of the multicast communication problem.

An intuitive and trivial solution to this problem is to rebuild the tree using a static algorithm whenever a join request by a new member, or a leave request by a current member, must be handled by the network group management protocol. However, such a solution may have repercussions for members who remain in the group since there may be a disturbance in the communication.

Furthermore, such a change may cause packets to arrive out of order. Another solution is to permit only local or partial reconfigurations when modification to the group membership are required. Yet another approach is to start with an optimal tree and make minimal changes as group membership changes without causing disruption to the members who remain in the group. This approach, however, may not be as efficient as the other approaches because no reconfiguration of the current tree is allowed.

Waxman was one of the first researchers to address the online multicast tree problem [17, 9]. In his work, Waxman partitioned the on-line multicast heuristics into two types: the ones that do not allow rearrangement (*nonrearrangeable*) of the multicast tree and those that allow rearrangement (*rearrangeable*) when the cost exceeds some limit. Several heuristics which approximate the OMCMT problem have been suggested [17, 4, 6, 10, 11, 18, 9], but none of these address supporting the delay requirements of multimedia applications. Following is a review of some of these heuristics.

#### (a) On-Line Greedy Heuristic (OGH)

This heuristic works as follows. In response to a join request, a node is added to the multicast tree using the shortest path from the current multicast tree to that node. For each leave request, the node is marked as a non-multicast node and is deleted only if it is a leaf node. This is achieved by removing the leaf node from the multicast tree and all branches linking that node to the tree [17]. Imase and Waxman proved that in the case where only node additions are allowed, the worst case cost scenario of the multicast tree produced by OGH is no worse than twice the cost of the multicast tree produced by the best nonrearrangeable algorithm for the online Steiner tree [9].

#### (b) Edge Bounded Algorithm

Edge Bounded Algorithm (EBA) is a rearrangeable algorithm in which a partial rearrangement is permitted when a modification to the membership occurs. EBA bounds the worst case performance of the generated tree to  $4\alpha$  times that of an optimal Steiner tree, where  $\alpha$  is a constant value [9]. Also, it limits the number of rearrangements to  $O(K^{3/2})$  where  $K$  is the number of (join, leave) requests served.

The algorithm works by creating distance graphs  $G'$  and  $T'$ .  $G'$  is a graph derived from the original graph  $G$ . The nodes of  $G'$  are those of  $G$ . The edges of  $G'$ , however, are built in a way such that  $G'$  is a complete graph. Furthermore, the weights of  $G'$ 's edges are the costs of the minimum cost paths between the nodes of  $G$ . A multicast tree  $T'$  is created for the node set  $Z$  ( $Z = \{s\} \cup D$ ) from  $G'$  by pruning the minimum spanning tree of  $G'$ . For each join request, EBA selects the least cost path from the new node  $v$  to the closest node  $u$  in  $T'$ . EBA verifies that the added path is  $\alpha$  bounded by ensuring that the cost of the maximum cost edge on the path between  $v$  and any node  $u$  in  $T'$  does

not exceed  $\alpha$  times the cost of edge  $(v, u)$  in  $G'$ . If a path to a node  $u$  exceeds that limit,  $v$  and  $u$  will be connected using the least cost path.

Based on this algorithm, a delete request issued by a node  $v$  is handled in a way that depends on the degree of  $v$ . If the degree of  $v$  is one,  $v$  is removed using a procedure similar the one described in OGH. If the node has degree three or more, the node will be marked as deleted and no action will be taken. If the degree is two then the node will be removed along with the two adjacent edges. That will create two subtrees. These two subtrees will be connected using an edge that minimizes the cost of the path between the two subtrees.

After serving any join or leave request, EBA verifies that  $T'$  is still an extension tree. An extension tree is a tree in which the degree of any non-multicast member is greater than two. If the degree of a non-multicast node is two, the same process used to remove a node whose degree is two is undertaken to build an extension tree.

### (c) Shortest Path Tree

Doar and Leslie suggested adding a node by using the shortest path from the source to that node [6]. Thus, the multicast tree will be the union of the minimum source-to-destination shortest paths. Such a tree will give the same result whether the tree was built dynamically or statically. As a result, this approach makes the process of building multicast trees less prone to major spikes of inefficiency. Furthermore, the algorithm does not require handling of rearrangements when nodes join or leave the session. Doar and Leslie showed that such a tree is on average more than 60% worse than the optimal multicast tree. They also suggested imposing a hierarchal model which emulates a real network architecture composed of major backbones and subnetworks. With such a model they showed, by simulation, that the resulting trees are on average less costly than trees produced by non-hierarchal model because there is more sharing of the backbone links.

### (d) The Geographic-Spread Dynamic Multicast Heuristic (GSDM)

GSDM is a rearrangement heuristic which was proposed by Kadirire [11]. It is an optimization of the OGH. To illustrate this process, assume that a given node  $A$  issued a request to join the multicast tree. Furthermore, assume that node  $B$  is the closest tree node to node  $A$ , and nodes  $C$  and  $D$  are the two closest nodes to  $B$ . Based on this configuration, the heuristic selects the least cost path among  $C-B-D$  &  $B-A$ ,  $C-B-A-D$ , and  $C-A-B-D$ . If more than one minimum cost path exists, the heuristic selects the path that maximizes the *Geographic Spread*(GS) of the resulting tree [10].

The GS is defined as follows: let  $T$  be a tree that spans  $Z$  ( $Z = \{s\} \cup D$ ) where  $Z \subseteq V$  and let  $v \in V$  and  $z \in Z$ , the GS is defined as the inverse of the sum of the minimum distance from  $v$  to all  $z \in Z$  for all nodes  $v \in V$  as

shown in equation 1. It has been shown that GSDM heuristic usually performs slightly better than OGH [11].

$$GS(T) = \left[ \sum_{v \in V \& z \in T} SP(v, z) \right]^{-1} \quad (1)$$

where  $SP(v, z)$  is the shortest path between  $v$  and  $z$ .

### (e) ARIES

ARIES (A Rearrangeable Inexpensive Edge-Based On-Line Steiner Algorithm) is a rearrangeable heuristic [4]. ARIES performs a rearrangement of a region of the multicast tree when the number of modifications (join, leave) within that region reaches a threshold. A region is defined as the part of the multicast tree whose interior nodes are non-stable nodes. A stable node is a node that has never been modified since the start of the tree or the last rearrangement of that region. The performance and the time complexity of ARIES algorithm depend on the threshold value. Small threshold values improve ARIES's performance and increase its run time, and vice versa.

## 3 HEURISTIC SELDOM

The objective of the OMCMT problem is two fold: minimizing the multicast tree cost and bounding the delays from the source to the destinations. Minimizing the cost by itself is a harder problem since the problem reduces to the Steiner tree problem which is inherently NP-complete. On the other hand, finding a multicast tree which satisfies the specified delay requirements can be solved in polynomial time using one of the classical minimum cost path algorithms [5]. When the two parameters are combined together, the problem is still NP-complete, even in the static case.

On-line multicast algorithms must handle dynamic group membership requests to *join* or *leave* a multicast session in progress. These requests require updating the multicast tree by adding the joining node or removing the leaving node from the tree in a way such that the overall cost of the tree remains minimal and the delay bounds of all multicast nodes are still satisfied. This online version of the multicast problem is still NP-Complete both in a rearrangeable and nonrearrangeable setting. Therefore, some heuristic that gives a "good" approximation with low run time overhead is needed. Furthermore, the heuristic must avoid disrupting the current connections and cause a minimal amount of change to the current connections. More specifically, in the case of joining request, the objective of an online multicast algorithm is to find a low-cost, delay-bounded path to the new joining node, while maintaining the lowest possible number of arrangements. Furthermore, in the case of leave request, the algorithm should delete a leaving node in a way such that the

delay bounds are not violated and the number of arrangements remains low. In the following, we first formulate the online multicast problem. We then propose a new online heuristic, referred to as *Simple and Efficient, Low-Cost, Delay-Bounded Online Multicasting* (SELDOM), which provides an effective solution to handle join and leave requests efficiently.

### 3.1 Problem Definition

A point-to-point communication network can be represented as a directed graph  $G = (V, E)$  where  $V$  denotes a set of nodes and  $E$  a set of asymmetric links. The network is assumed to be full duplex. In other words, the existence of link  $l = (u, v) \in E$  implies the existence of link  $l' = (v, u) \in E$ . Each link  $l \in E$  is assigned a cost value  $C : E \rightarrow \mathcal{R}^+$ . A link cost value can be either the link utilization or a monetary value associated with the link. Also, each link  $l \in E$  has delay  $\delta(l)$ , where  $\delta(l) \in \mathcal{R}^+$ . A link delay may consist of CPU processing, queuing, transmission and propagation. Because network links are often asymmetric, it is often the case that  $C(u, v) \neq C(v, u)$  and  $\delta(u, v) \neq \delta(v, u)$ . If the links are symmetric, then  $C(u, v) = C(v, u)$  and  $\delta(u, v) = \delta(v, u)$ .

A path from node  $v_1$  to  $v_k$  is defined as the sequence of nodes and links  $P(v_1, v_k) = v_1, v_2, \dots, v_k$  such that  $(v_i, v_{i+1}) \in E$  for all nodes from  $v_1$  to  $v_{k-1}$ . The path  $P(v_1, v_k)$  is assumed to be loop free. The cost of a path is the sum of the cost of the links constituting  $P(v_1, v_k)$ :

$$Cost(P(v_1, v_k)) = \sum_{l \in P(v_1, v_k)} C(l) \quad (2)$$

Similarly, the total delay of a path is the sum of the delay of the links constituting  $P(v_1, v_k)$ :

$$Delay(P(v_1, v_k)) = \sum_{l \in P(v_1, v_k)} \delta(l) \quad (3)$$

The general MCMT problem can be defined as follows: let the multicast group consist of a source node  $s$  and a destination node set  $D$ . Given that the maximum delay allowed on the path  $P(s, d)$ , where  $d \in D$ , is  $\Delta_d$ , the MCMT,  $T$ , is the tree that satisfies the following two conditions:

$$Cost(T) = \sum_{l \in T} C(l) \quad \text{is minimum} \quad (4)$$

subject to

$$Delay(P(s, d)) = \sum_{l \in P(s, d)} \delta(l) < \Delta_d \quad \forall d \in D \quad (5)$$

In some cases, the underlying multicast application is delay insensitive. In this case, the delay is not bounded ( $Delay(P(s, d)) = \infty, \forall d \in D$ ), and the MCMT problem reduces to minimizing the total cost of the multicast tree. Furthermore, if the links are undirected, the MCMT is reduced to finding the minimum cost tree,  $T$ , that contains all nodes  $\{s\} \cup D = Z$ . This problem is well known in the literature and is called Steiner Minimal Tree (SMT) [8].

The above definition applies to the case where the multicast nodes are known a priori (i.e. off-line multicasting). However, when the problem is on-line, the multicast tree is dynamic; the multicast nodes in this case may join or leave dynamically. This problem is called the *Online Minimum Cost Multicast Tree* (OMCMT) problem. In this case, a request vector  $R = (r_1, r_2, \dots, r_k)$  with  $k$  requests is given where  $r_i$  has three parameters  $(v, x, \Delta_v)$  such that  $v \in D$ ,  $x \in \{add, remove\}$  and  $\Delta_v$  is the delay bound to that node. The OMCMT tree in this case is defined as the tree that satisfies the two conditions in equations 4, 5 after processing each join or leave request.

### 3.2 SELDOM Design Approach

Given that some of the networks are not always congested and the number of delay violations may be low, the approach taken by SELDOM to efficiently handle online multicast requests is to first minimize the cost of the paths to destinations, and second verify the feasibility of these paths in supporting the required delay bounds. This approach is being taken since the cost reduction is inherently an NP-complete problem while bounding the delay is a simpler problem which can be performed in polynomial time.

In response to a join request, SELDOM determines the least cost path from the multicast tree to the new node, and verifies the feasibility of the selected path in meeting the delay bound requirements of the joining node. If the delay requirements of the new added node are violated, SELDOM searches for a delay-bounded path with the lowest cost. Following is a description of two modes of SELDOM operations, namely nonrearrangeable SELDOM, and rearrangeable SELDOM.

### 3.3 Nonrearrangeable SELDOM

In a connection oriented network, it is important to reduce the number of rearrangements of current connections as a multicast tree evolves. This is

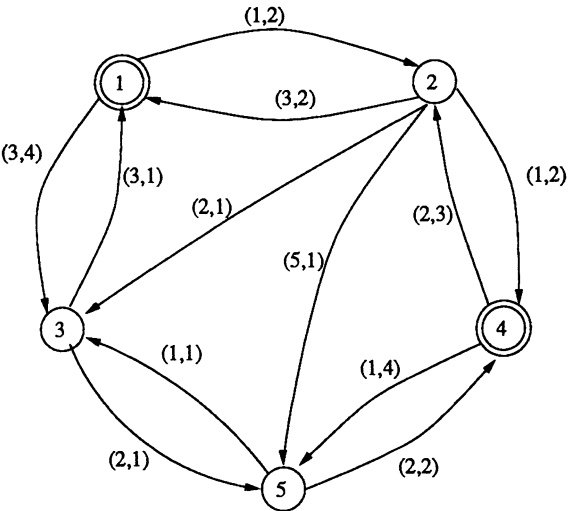
especially important when the network is loaded and the network control and routing information is distributed. Rearrangement involves rerouting of information and may require a significant amount of network synchronization and resources. Therefore, a nonrearrangeable online multicast algorithm is desirable when rearrangement of the multicast tree is difficult.

First, the SELDOM nonrearrangeable mode is presented. In this mode, SELDOM does not require any rearrangement of the multicast connections. The nonrearrangeable mode of SELDOM works as follows: for each incoming request, if it is a leave request, the node is marked as a non-multicast node and it is deleted only if it is a leaf node. The deletion is achieved by removing the leaf node from the multicast tree and all branches linking that node to the tree. If it is a join request then the following algorithm is used:

Let  $G$  be the network graph,  $T$  be the current multicast tree, and  $v$  be the node to be added.

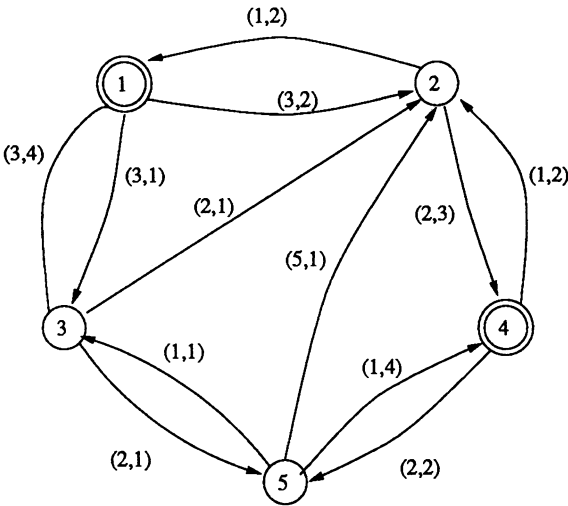
1. Find the least cost path  $SP(T,v)$  from  $T$  to  $v$ . If  $SP(T,v)$  total delay is bounded then return that path and quit. Otherwise, perform the following steps.
2. Create a new graph  $G'$  which consists of  $G$ 's nodes and  $G$ 's edges reversed.
3. Let  $P$  be the set of the shortest delay paths from  $v$  to  $T$ 's nodes in  $G'$ .
4. Remove the edges of  $T$ , from  $G'$ .
5. Find the set of the least cost paths from  $v$  to  $T$ 's nodes in  $G'$  and add them to  $P$ .
6. Out of  $P$ , pick the path  $p$  which satisfies the delay bound such that  $delay(p) < \Delta_v$  and  $cost(p)$  is minimum.
7. If no such path exists then return "cannot add this node".
8. Return  $p$ .

$G$ 's links are reversed to speed up the shortest path calculations. The edges of  $T$  are removed to create independent paths. To explain the above idea further and to show the benefit from removing  $T$  links, we give the following example. In Figure 1 the source node is node 1 and the current multicast tree includes multicast node 4 (in addition to the source node 1) and the maximum delay bound for all multicast nodes is 7. Initially the multicast tree consists of nodes 1, 2, and 4 with link (1,2) and (2,4). The cost of the tree is 2 and delay to node 4 is 4. Assume that a new request comes to add node 5 to the multicast tree. Using SELDOM, it will first try adding node 5 using the least cost path from  $T$  to 5 which is path (4,5). However, the delay on that path is 8 which violates the delay bound. Therefore, SELDOM will try to find a better delay-bounded path. First, SELDOM will create a new graph  $G'$  which is similar to  $G$  but the links are reversed as show in Figure 2. The reversal of the links is performed to speed up the shortest path computation from the violating node to the tree. Then, SELDOM will compute the least delay paths in  $G$  from the  $T$  nodes to node 5. This can be performed in  $O(n^2)$  using the



**Figure 1** SELDOM example, node 1 is the source node.

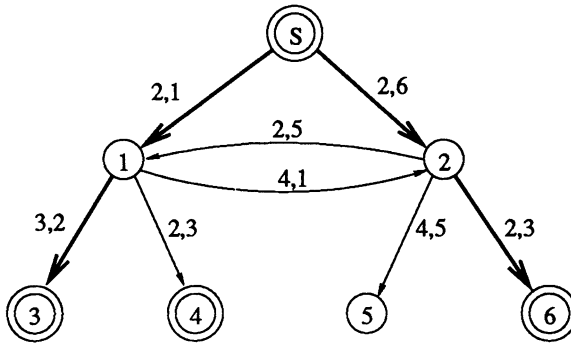
---



**Figure 2** SELDOM example after reversing G's links

---





**Figure 3** SELDOM example with a source node and two destinations, 2 and 6, each link is assign two values (cost and delay).

---

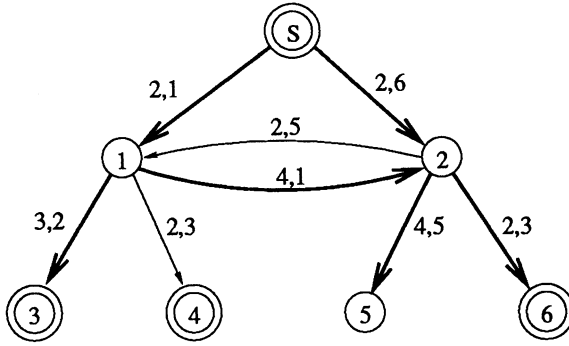
shortest delay paths in  $G'$ . These paths are: (5,2,1), (5,2), (5,4). After that, the  $T$  links ((1,2),(2,4)) are removed from  $G'$ . The least cost paths from node 5 to the  $T$  nodes will be computed. These paths are: (5,3,1), (5,3,2), and (5,4). Out of these six paths, path (5,3,2) gives the least-cost bounded path with additional cost of 4 and a total delay (from the source to node 5) of 4. If the  $T$  links were not removed, the least cost paths will be (5,4,2,1), (5,4,2) and (5,4). These paths are not independent because they use  $T$ 's links (1,2) and (2,4). Hence, without removing  $T$  links, path (5,3,2) would not be discovered.

### 3.4 Rearrangeable SELDOM

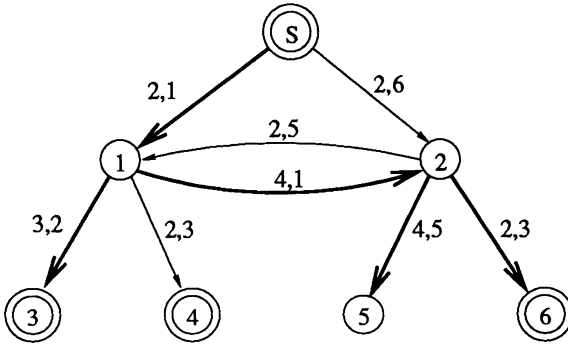
Handling join and leave requests in rearrangeable networks is more involved than in nonrearrangeable settings. In the following, we first describe the operations SELDOM undertakes to respond to a join request. We then describe the operations required to handle a leave request.

#### (a) Join Request

When adding a node to the current multicast tree in response to a join request, the nonrearrangeable SELDOM may produce a multicast graph which has a node with two incoming paths. As an example, consider the graph depicted in Figure 3. In this graph, the multicast set consists of the source node and two destination nodes, 2 and 6, with a delay bound of 10. The multicast tree in this case is marked using bold links with total cost of 9. Assume that node 5 wants to join the multicast set. In this case, node 5 can be added using the nonrearrangeable SELDOM heuristic. Path 2,5 cannot be used because its traffic will come through link S,2 with a path delay of 11 which violates the delay bound. In this case, node 5 will be added to the multicast tree using



**Figure 4** SELDOM example after adding node 5.



**Figure 5** SELDOM example after pruning the link from the source to node 2.

path 1,2,5 as shown in Figure 4 with a total tree cost of 17. This new path makes node 2 have two incoming paths. The generated multicast tree satisfies the delay requirements but this configuration makes node 2 do double work for the same packets.

A multicast tree with a node which has two incoming paths will deliver the multicast traffic as it should. However, that will increase routers' state information. In the above case, a router will have to remember what to do, although it handles the same packets. Also, that will double the workload of the router for the same packets because the router is forced to process the same packets twice. Furthermore, it will increase the cost of tree.

To reduce and possibly eliminate this extra overhead, one of the two incoming paths has to be pruned. Pruning a path, however, should be performed in a way such that no delay bounds to any of the multicast destinations are

violated. In order to achieve this, the path with the larger delay should be pruned. Thus, path S,2 will be pruned because it has a delay of 6 whereas path S,1,2 has a delay of 2 as shown in figure 5. The total cost of the multicast tree after pruning is 15.

**Lemma:** if the current multicast tree does not have a node with more than one incoming path, then when a node is added, the new path will not create a node with more than two incoming paths.

The above can be proved as follows. When a node is added, the shortest path from a tree node to the new node will not have a cycle because any shortest path cannot include cycles. Therefore, a path will not go through a node more than once. Consequently, the new path will not add more than one extra incoming path to any node in the multicast tree. Since no multicast node has more than one incoming path, the resulting multicast graph will never have a node with more than two incoming paths.

Based on the above we propose a rearrangeable mode of SELDOM. The node join process of this mode tries to reduce the overall cost by pruning the extra paths while minimizing the number of rearrangements. For every node addition it finds all possible paths as it is done in the the previous mode. Then, for each possible path that satisfies the delay bound, it computes the cost of the multicast tree by adding the cost of the new path minus the cost of any possible pruned paths that can exist if that node is added. More formally, the *node-join* algorithm is defined as follows:

1. Find the least cost path  $SP(T,v)$  from  $T$  to  $v$ . If  $SP(T,v)$  total delay is bounded then return that path and quit. Otherwise, perform the following steps.
2. Create a new graph  $G'$  which consists of  $G$ 's nodes and  $G$ 's edges reversed.
3. Let  $P$  be the set of the shortest delay paths from  $v$  to  $T$ 's nodes in  $G'$ .
4. Remove the edges of  $T$ , from  $G'$ .
5. Find the set of the least cost paths from  $v$  to  $T$ 's nodes in  $G'$  and add them to  $P$ .
6. For each possible path in  $P$ , compute the cost of resulted multicast tree including the new possible path. The cost of the tree is computed by finding the current cost of the multicast tree including the new path minus the cost of any possible pruned paths that can result from adding that path.
7. Out of  $P$ , pick the path  $p$  which satisfies the delay bound such that  $delay(p) < \Delta_v$  and the total  $cost(T)$  is minimum.
8. If no such path exists then return "cannot add this node".
9. Return  $p$ .

If all nodes that joined the multicast tree directly use the shortest cost paths without violating any delay bounds, then the produced tree will be similar to the tree produced by OGH (online Greedy Heuristic). It was shown by simulation that when there are only node additions, OGH produces trees

with an average cost that does not exceed that of the optimal tree cost by more than 10%. Imase and Waxman [9] proved that in the worst case the cost of the multicast tree produced by OGH in an undirected graph is no worse than twice the cost the multicast tree produced by the best nonrearrangeable algorithm.

### (b) Leave Request

In the nonrearrangeable mode, the deletion of a node in response to a leave request, causes SELDOM to mark the node as a non-multicast node. Furthermore, if the node is a leaf node, all edges and nodes in the relay path linking that node to the tree, including the leaving node itself, are removed from the tree. In the rearrangeable SELDOM, however, node deletion can be improved by performing limited arrangement. The basic steps undertaken by SELDOM to remove a node in response to a leave request are described below:

Assume the multicast tree receives a leave request from node  $v$ . Let  $\deg(v)$  be the degree of node  $v$ . Also, let a relay path be the path whose all internal nodes have degree two and they are not multicast members. Then the node deletion algorithm can be explained as follows:

If node  $\deg(v) > 2$  then

mark  $v$  as a non-multicast member.

else if  $\deg(v) = 2$  then

Delete the relay path from  $v$  up to node  $v_{left}$  and the relay path up to node  $v_{right}$  where  $v_{left}$  is the end node of the relay path on the left of  $v$  and  $v_{right}$  is the end node of the relay path on the right of  $v$ . The above will divide the multicast tree into two subtrees.

Assume that the source node is in  $v_{left}$  subtree. Also, assume  $T_1$  is  $v_{right}$  subtree. Reconnect node  $v_{left}$  and  $T_1$  using the least cost path from  $v_{left}$  to  $T_1$  which does not violate the delay bound. The least cost path is the lowest-cost, delay-bounded path among the following paths: the least-cost paths from  $v_{left}$  to every  $T_1$  node, the least delay paths from  $v_{left}$  to every  $T_1$  node, and the old path between  $v_{left}$  and  $v_{right}$ .

else

Delete  $v$  and its relay path up to  $u$  where  $u$  is the end node of the relay path. If  $u$  is not a multicast member then delete  $u$  along with its two relay paths up to  $u_{left}$  and  $u_{right}$ . Assume that the source node is in  $u_{left}$  subtree and  $T_1$  is the subtree of  $u_{right}$ . Reconnect  $u_{left}$  and  $T_1$  in a way similar to connecting  $v_{left}$  and  $T_1$  in the above.

The above deletion algorithm limits the number of rearrangements to one. Because it requires finding the shortest path tree, the time complexity of the algorithm is  $O(n^2)$ . The above enhanced deletion algorithm is expected to reduce the cost of the multicast tree. However, the algorithm can be improved even further. To illustrate this improvement, assume that  $T_1$  is the subtree that includes the end node of the left relay path and  $T_2$  is the subtree that includes the end node of right relay path. Then, a better path is the one that connects  $T_1$  subtree with  $T_2$  subtree instead of the best path that connects one node with the other subtree. However, that makes the algorithm more complex because it requires finding the least cost paths from every  $T_1$  node to every  $T_2$  node. This process has a time complexity of  $O(n^3)$ .

Because SELDOM uses the shortest delay paths, it finds a solution if one exists. The time complexity of a node addition is similar to the time complexity of the shortest path algorithm which is  $O(n^2)$  where  $n$  is the number of nodes in the graph. That is because in the worst case it requires computing the cost of the least-delay paths and the least-cost paths from the multicast nodes to the node being added. By reversing the direction of the links, this still can be done in  $O(n^2)$ . The complexity of link pruning will never exceed the number of links. Similarly, the time complexity of node deletion is bounded by  $O(n^2)$ . Hence, the overall time complexity of SELDOM is  $O(n^2)$  for each node addition or deletion.

## 4 CONCLUSION

The multicast problem is NP-complete. Therefore, some heuristics were suggested to give a good approximation with polynomial time complexity. This paper started with a discussion of low cost, delay-bounded multicast trees. Next the online multicast problem was discussed. The online multicast problem is difficult because members join and leave the multicast tree dynamically. One possible solution is to use any of the static multicast heuristics to solve the online multicast problem. However, that will be costly because it requires tearing down and reconnecting the current multicast tree connections. A few heuristics for the online problem were presented. Some of these heuristics did not require rearrangements while the others tried to limit the number of rearrangements. The heuristics that allow rearrangement, however, usually give better results. All of these suggested online heuristics do not bound the delay.

A new heuristic, SELDOM, for online, low-cost multicasting for real-time applications was presented. The nonrearrangeable mode of SELDOM adds a node using the shortest path if that path does not violate the delay bound. If the path violates the delay bound, a search for a low-cost, delay-bounded path is performed. A node is deleted by removing the node and the relay path from the tree to that node. A rearrangeable mode of SELDOM improves the joining process by pruning any possible two incoming paths. Also, it enhances the node leaving process by making limited rearrangement to the graph if that

node has a degree of two. The time complexity of both the nonrearrangeable and the rearrangeable SELDOM is  $O(n^2)$ .

## REFERENCES

- [1] Tawfig Alrabiah and Taieb Znati. Low-cost, bounded-delay multicast routing for qos-based networks. *Technical Report*, TR-98-3, November 1997.
- [2] Tawfig Alrabiah and Taieb Znati. A simulation framework for the analysis of multicast tree algorithms. *The 30th Annual Simulation Symposium*, 30:196–205, April 1997.
- [3] T. Ballardie, P. Francis, and J. Crowcroft. Core-based trees (cgt) an architecture for scalable inter-domain multicast routing. *Computer Communication Review*, 23(4):85–95, 1993.
- [4] F. Bauer and A. Varma. Aries: A rearrangeable inexpensive edge-based on-line steiner algorithm. *IEEE Journal on Selected Areas in Communications*, 15(3):382–397, April 1997.
- [5] E. W. Dijkstra. A note on two problems in connection with graphs. *Numeriske Mathematik*, 1:269–271, 1959.
- [6] Matthew Doar and Ian Leslie. How bad is naive multicast routing. *Proceedings of IEEE INFOCOM*, San Francisco, CA:82–93, April 1993.
- [7] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. Protocol independent multicast-sparse mode (PIM-SM): Protocol specification. *Internet RFC 2117*, <http://ds.internic.net/rfc/rfc2117.txt>, June 1997.
- [8] S. L. Hakimi. Steiner's problem in graphs and its implications. *Networks*, 1:113–133, November 1971.
- [9] Makoto Imase and Bernard Waxman. Dynamic steiner tree problem. *SIAM Journal of Discrete Math.*, 4(3):369–384, August 1991.
- [10] James Kadirire. Minimizing packet copies in multicast routing by exploiting geographic spread. *SIGCOMM Communication Review*, 24(3):47–62, July 1994.
- [11] James Kadirire and Graham Knight. Comparison of dynamic multicast routing algorithms for wide-area packet switched (asynchronous transfer mode) networks. *IEEE INFOCOM, Boston*, 19:212–218, April 1995.
- [12] V. Kompella, J. Pasquale, and G. Polyzos. Multicast routing for multimedia communication. *IEEE/ACM Transactions on Networking*, 1(3):286–292, June 1993.
- [13] J. Moy. Multicast routing extensions for OSPF. *Communications of the ACM*, 37(8):61–66, August 1994.
- [14] T. Pusateri. Distance vector multicast routing protocol. *Internet Draft*, February 1997.
- [15] Q. Sun and H. Langendoerfer. Efficient multicast routing for delay sen-

- sitive applications. *Proceedings of Second Workshop Protocols Multimedia Systems (PROMS'95)*, pages 452–458, April 1995.
- [16] H. Takahashi and A. Matsuyama. An approximate solution for the steiner problem in graphs. *Mathematica Japonica*, 24:573–577, 1980.
  - [17] Bernard M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1611–1622, December 1988.
  - [18] Bernard M. Waxman. Performance evaluation of multipoint routing algorithms. *IEEE INFOCOM*, pages 980–986, 1993.
  - [19] Pawel Winter. Steiner problem in networks: A survey. *Networks*, 17:129–167, 1987.
  - [20] Pawel Winter and J. MacGregor Smith. Path-distance heuristics for the steiner problem in undirected networks. *Algorithmica*, 7:309–327, 1992.
  - [21] Q. Zhu, Mehrdad Parsa, and J. Garcia-Luna-Aceves. A source-based algorithm for delay-constrained minimum-cost multicasting. In *Proc. IEEE INFOCOM 95*, pages 377–385, 1995.

## 5 BIOGRAPHY

Tawfig F. Alrabiah received B.A. degree in quantitative methods from King Saud University, Saudi Arabia and the M.S. degree in Computer Science from the University of Pittsburgh in 1995. He also holds an M.S. degree in Information Science from the University of Pittsburgh. He is currently a Ph.D candidate at the University of Pittsburgh majoring in computer Science.

His current research interests are in developing network routing and path establishment schemes to support unicast and multicast communication in distributed multimedia systems. Mr. Alrabiah is a member of ACM and IEEE.

Taieb Znati is an Associate Professor of Computer Science at the University of Pittsburgh, with joint appointments in Telecommunications (Department of Information Science) and Computer Engineering (Department of Electrical Engineering).

Dr. Znati current research interests are in the areas of wired and wireless real-time communication networks, with a particular emphasis on the design and analysis of communication protocols to support distributed multimedia applications. He has published numerous papers in these areas and developed different frameworks to support quality of service requirements of multimedia applications. Dr. Znati chaired multiple conferences and workshops in the field of distributed multimedia systems, high speed communication networks and simulation. He is the current General Chair of the Annual Simulation Symposium, and the General Chair of the Communication Networks and Distributed Systems Modeling and Simulation Conference. He frequently participates in panel discussions about future developments in multimedia systems and high speed networks from the perspective of researchers, developers and users.

# **Part Three**

---

## **Scalable Multicast**



# A Scalable and Robust Feedback Mechanism for Adaptive Multimedia Multicast Systems

*Alaa Youssef\*, Hussein Abdel-Wahab, and Kurt Maly*  
*Department of Computer Science*  
*Old Dominion University*  
*Norfolk, VA 23529, USA*  
(youssef,wahab,maly)@cs.odu.edu

## Abstract

We present a mechanism for providing state feedback information to multicast sources of multimedia streams in a scalable and robust manner. The presented feedback mechanism is suitable for best-effort unreliable networks such as the Internet. This mechanism is useful for controlling the transmission rate of multimedia sources in both cases of layered and single-rate multicast. It allows for determining the worst case state among a group of receivers, where each receiver may be in one of a set of finite states, and is applicable in receiver-driven as well as in sender-driven adaptive multimedia systems. Simulation results show that the presented feedback mechanism scales well for very large groups of up to few thousands of participants. The efficiency of the proposed mechanism in eliminating the reply implosion problem, its robustness in facing network losses, as well as its responsiveness are illustrated. In addition, the advantages of the proposed mechanism over other feedback mechanisms are demonstrated. Moreover, adaptive enhancements for the mechanism are proposed to maintain its scalability for even larger groups.

## Keywords

Feedback, multicast, adaptive multimedia applications.

## 1 INTRODUCTION

Multimedia streams are becoming a main component of modern distributed collaboration and tele-teaching systems. Most of these systems rely on IP multicasting in order to scale to large groups of participants. However, the quality of service (QoS) requirements of the multimedia streams demand special treatment. The main approaches taken for handling the requirements of

---

\*This work is supported in part by IBM.

multimedia streams can be broadly classified as either proactive or reactive. The proactive approach relies on the existence of a resource reservation protocol (Gupta *et al.* 1995, Zhang *et al.* 1993), and underlying scheduling mechanisms, to reserve and guarantee end-to-end resources. On the other hand, the reactive approach relies mainly on the ability of the application to adapt itself to the level of available resources (Bolla *et al.* 1997, Bolot *et al.* 1994, Cheung *et al.* 1996, McCanne *et al.* 1996). Most of these approaches, for handling multimedia streams, manage individual connections in isolation of others, which may lead to a state of competition for resources among streams belonging to the same session, thus decreasing the overall perceived session quality. Our approach, however, is to dynamically control the QoS offered by the system across the set of connections belonging to the application. This control is based on the application semantics, and focuses on maintaining the best overall quality of session, at every instant during the session lifetime. To this end, we introduced the concept of *Quality of Session (QoSess)* (Youssef *et al.* 1997).

In (Youssef *et al.* 1998), we propose an architecture for a middle-ware platform, which supports collaborative multimedia applications by providing QoSess control mechanisms. Conceptually the QoSess control layer acts as a closed loop feedback system that constantly monitors the observed behavior of the streams, takes inter-stream adaptation decisions, and sets the new operating level for each stream from within its range of permissible operating points. Over wide area network connections, the QoSess control layer manages the resources that are collectively reserved, for the streams of a distributed application, by a resource reservation protocol, such as RSVP. Multi-grade streams are centric to the QoSess framework, in order to support heterogeneity of receivers and network connections. Multi-grade transmission can be achieved either by hierarchical encoding (McCanne *et al.* 1997, Sengel *et al.* 1997), or by simulcast which is the parallel transmission of several streams each carrying the same information encoded at a different grade (Li *et al.* 1996, Willebeek-LeMair *et al.* 1997).

In this paper, we present one of the main building blocks of the QoSess control layer: a scalable and robust state feedback mechanism. This mechanism provides the source of a multimedia stream with deterministic information regarding the state of the receivers. The state of a receiver may be defined as the layers which it is interested in receiving from the source of a hierarchically encoded stream. Given this knowledge, the sender can suppress or start sending the correct layers. The feedback mechanism is not only important for saving the sender's host and LAN resources but for saving WAN resources as well in situations where the application's addressing scheme for the layers does not permit the intermediate routers to suppress unwanted layers, or where the session is conducted over an Intranet whose subnets are inter-connected via low level switches that do not implement the IGMP protocol (Deering *et al.* 1990) for suppressing unwanted multicast packets. Soliciting feedback

from receivers in a multicast group might create a *reply implosion* problem, in which a potentially large number of receivers send almost simultaneous redundant replies. We present a scalable and robust solution to this problem.

The rest of this paper is organized as follows. In Section 2, the role of feedback in several adaptive multimedia multicast systems is illustrated. A brief survey of the different approaches for providing scalable feedback is presented in Section 3. The proposed feedback mechanism is described in detail in Section 4, followed by a performance study and comparison in Section 5. In Section 6, adaptive enhancements for the proposed mechanism in order to support very large groups of receivers are described, and we present our conclusions in Section 7.

## 2 FEEDBACK ROLE IN MULTIMEDIA MULTICAST

Early attempts towards providing adaptive transport of multimedia streams over the Internet focused on the sender as the entity playing the major role in the adaptation process (Bolot *et al.* 1994, Busse *et al.* 1995). Information about the congestion state of the network, as seen by the receivers, was fed-back to the sender which used it to adapt to changes in the network state. In many cases, the monitored performance parameters (e.g., loss rate, delay, jitter, throughput) were mapped, by the receiver, to one of several qualitative performance levels, and reported to the sender (Bolot *et al.* 1994, Busse *et al.* 1995, Cheung *et al.* 1996). The sender adapted its transmission rate by varying the quality of the transmitted media content by means of controlling several encoder parameters (e.g., frame rate, frame size, or quantization step for video streams). The sender often based its decisions on the worst case state reported (Busse *et al.* 1995), and sometimes based it on a threshold of the number of receivers suffering the worst state (Bolot *et al.* 1994). In this approach all receivers have to receive the same quality of multimedia streams regardless of the differences in their capacities and the capacities of the network connections leading to them. Although sometimes it is desired to maintain identical stream quality across all participants of a session (e.g., for some discrete media streams), yet this is not always the case especially with continuous media streams.

The first approach, to address the need for providing a multi-grade service to participants of the same session, was represented by the introduction of the concept of *simulcast* (Li *et al.* 1996, Willebeek-LeMair *et al.* 1997). In a simulcast system, the sender simultaneously multicasts several parallel streams corresponding to the same source, but each is encoded at a different quality level. Each receiver joins the multicast group that matches its capabilities. Within a group, the same techniques of source adaptation, that were mentioned above, are applied within a limited range. Thus, the same feedback mechanisms are also deployed within each group.

With the advent of hierarchical encoding techniques (McCanne *et al.* 1997,

Senbel *et al.* 1997), a new trend in adaptive multimedia transport appeared in which the receiver plays the sole role in adaptation (McCanne *et al.* 1996). In such systems the receiver is responsible for determining its own capabilities, and consequently, it selects the number of layers to receive from the hierarchically encoded stream. The source, however, is assumed to be constantly multicasting all the layers.

While it is very obvious that the layered encoding approach is more efficient in the utilization of resources relative to the simulcast approach, yet it is still debatable whether layered encoding techniques will be able to provide the same media quality as the simulcast encoders which operate in parallel, each optimized for a particular target rate. In spite of this debate, the layered approach is the most appealing from the networking point of view, due to its efficient utilization of network resources, especially bandwidth. However, this approach as described is not as efficient as can be. The fact that the source keeps sending at full rate, all layers, constantly, may lead to the waste of more resources than with simulcast, in the case where no receiver subscribes to some of the layers. On the other hand, augmenting this approach with a simple scalable feedback mechanism that provides the source with information regarding which layers are being consumed and which are not, yields more efficiency in resource consumption, as the sender can get actively involved in the adaptation process by suppressing the unused layers.

The introduction of such a feedback mechanism, for receiver-oriented layered transport of multimedia streams, is not only an added efficiency feature for such transport protocols, but it is also a critical feature for the success of collaborative multimedia sessions in which multiple streams are concurrently active. In such collaboration sessions, multiple streams are typically distributed to all participants of the session, and the overall session quality is determined by the quality of each of the streams as well as by their relative importance and contribution to the on-going activity. In presence of scarce resources, it is logical to sacrifice the quality of one low priority stream for the sake of releasing resources to be used by a higher priority stream. Should the low priority stream source keep pushing all unused layers to the network, the decision taken by the receivers to drop these layers for releasing resources is rendered almost useless. This uselessness will hold true forever for the sender's host and LAN, while the rest of the network may eventually have these resources released as the multicast routers stop forwarding the unused layers. In situations where the application's addressing scheme for the layers does not permit the intermediate routers to suppress unwanted layers, WAN resources may also be wasted.

In the former case, besides the unnecessary delay in releasing resources, the fact that the sender's host and LAN will always be overloaded is very critical, as the session participants on this LAN may not be able to receive other higher priority streams. The problem is more crucial for Intranet based

collaboration systems since all the session participants (senders and receivers) are typically within a few hops from one another (Maly *et al.* 1997).

Moreover, since the sender may be sending only a subset of its layers, it needs to know about the existence of clients for higher layers that are currently suppressed, as soon as these clients subscribe to these layers. This information must be provided to the sender in a timely and scalable way that avoids potential implosion problems in such cases when many clients subscribe to higher layers almost simultaneously. This is likely to happen when some streams are shutdown releasing resources that can be utilized by other active streams.

From the above we conclude that a feedback mechanism is necessary for involving the sender in the adaptation process for receiver driven layered multicast of multimedia streams, especially in the context of collaborative multimedia sessions. Moreover, such a feedback mechanism is essentially the same as, and can replace, feedback mechanisms for supporting simulcast and single-rate multicasts. In the following section, we briefly describe the different approaches to providing scalable feedback, then in Section 4, we introduce the proposed scalable and robust mechanism for providing feedback in adaptive multimedia multicast systems.

### 3 EXISTING SCALABLE FEEDBACK TECHNIQUES

Soliciting information from receivers in a multicast group might create a *reply implosion* problem, in which a potentially large number of receivers send almost simultaneous feedback messages that contain redundant information. Typical solutions to address this problem include *probabilistic reply*, *expanding scope search*, *statistical probing*, and *randomly delayed replies* (Bolot *et al.* 1994).

**Probabilistic reply:** In a probabilistic reply scheme, a receiver responds to a probe from the source with a certain probability. If the source does not receive a reply within a certain timeout period, it sends another probe. This scheme is easy to implement. However, the source is not guaranteed to receive the worst news from the group within a certain limited period. In addition, the relationship between the reply probability and the group size is not well defined.

**Expanding scope search:** In the expanding scope search scheme, the time-to-live (TTL) of the probe packets sent by the source is gradually increased. This scheme aims at pacing the replies according to the source capacity of handling them, since the source does not re-send the probe with increased scope until it has processed all previous replies. Clearly this is efficient only in the case where the receivers are uniformly distributed in TTL bands, which may not be the case.

**Statistical probing:** This scheme relies on probabilistic arguments for scalability. At the start of a round of probes (called *epoch*), the sender and each of the receivers generate a random key of a fixed bit length. In each probe, the source sends out its key together with a number specifying how many of the key digits are significant. Initially, all digits are significant. If a match occurs at a receiver then that receiver is allowed to send a response. If no response is received within a timeout period, the number of significant digits is decreased by one and another probe is sent. In (Bolot *et al.* 1994), it was shown that there is a statistical relationship between the group size and the average round upon which a receiver first matches the key. This scheme is efficient in terms of number of replies needed to estimate the group size. However, as shown in (Bolot *et al.* 1994), the maximum response time (the time needed for the source to identify the worst case of all receivers) is equal to 32 times the worst case round trip time. For a typical worst case RTT of 500 milliseconds, it may take up to 16 seconds to find the worst case state of all receivers.

**Randomly delayed replies:** In the randomly delayed replies scheme, each receiver delays the time at which it sends its response back to the source by some random amount of time. Clearly, the success of this scheme in preventing the *reply implosion* problem depends to a great extent on the duration of the period from which random delays are chosen. However, the scheme is very appealing, in the sense that it allows for receiving responses from all the receivers in the group, if the delay can be adapted using some knowledge of the size of the group.

From the above basic mechanisms, the *randomly delayed replies* approach, augmented with suppression of redundant replies and careful selection of delay periods, is the most appealing for two main reasons: first, a response is always guaranteed; and second, the response time is expected to be always low. This is the basic idea deployed in IGMP (Internet Group Management Protocol) (Deering *et al.* 1990). In IGMP, the probe is sent to a local area network (LAN), and hence as soon as one of the receivers responds to the probe it is guaranteed that all the other receivers will hear that response and suppress their replies. Also, in such a local environment, the timeout period can be set to a fixed small value. In contrast, in our case, the group of receivers may be distributed over a wide area network (WAN), thus a reply sent by one receiver may not be heard by another before the other one emits its own reply which may be redundant. This implies the need for careful selection of the delay randomizing functions.

A closely related, but different, problem is the negative acknowledgment (NAK) implosion problem associated with reliable multicasting. A solution for the NAK implosion problem, which is based on randomly delayed replies with suppression of redundant NAKs, is adopted by the SRM protocol (Floyd *et al.* 1995). In SRM, when a receiver detects a lost packet, it randomizes

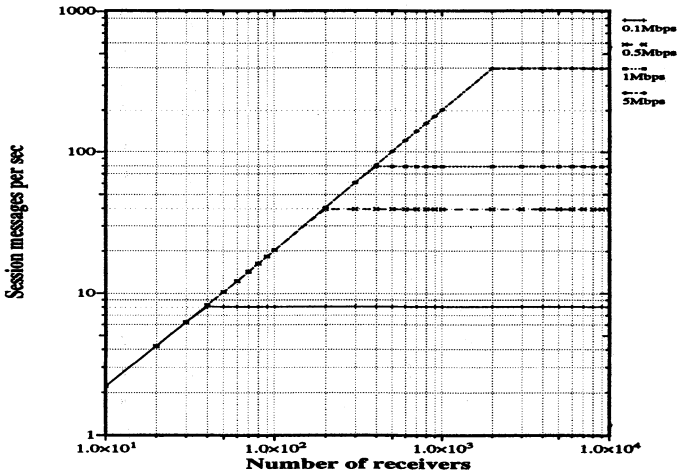


Figure 1 Overhead of session messages

the delay before sending its NAK in the interval  $[C_1 d_i, (C_1 + C_2) d_i]$ , where  $d_i$  is the distance from receiver  $i$  to the source,  $C_1$  and  $C_2$  are constant parameters. Both the NAK and the state feedback implosion problems are similar in the need for soliciting replies from a potentially very large group of receivers. However, with NAKs, whenever a data packet is lost on a link, all the receivers that the faulty link lead to will eventually detect the loss and send a NAK. Thus the distance between a receiver and the faulty link is the major factor that determines when the receiver will detect the fault, and consequently favoring closer receivers, by letting them send their NAKs earlier, implies suppression of more redundant NAKs. On the other hand, in the state feedback problem, the capacity of the receiver, and consequently its state, may not be related to its distance from the source. Therefore, a different criteria for randomizing the delays is required.

In SRM, each receiver must determine its distance from the source to use it in the delay function. The overhead of session messages (typically RTCP reports (Schulzrinne *et al.* 1996)) which are needed for that is not negligible. Figure 1, shows the overhead of RTCP reports for different session sizes and rates, assuming a single source. One of the objectives of the proposed feedback mechanism is to eliminate this high overhead, by designing the mechanism in a way that is not dependent on periodic session messages.

## 4 A SCALABLE FEEDBACK MECHANISM

In this section, we describe the proposed mechanism for eliciting feedback information from the receivers in a multicast group. The objective of the algorithm is to find out the worst case state among a group of receivers. The definition of the worst case state is dependent upon the context in which the feedback mechanism is applied. It can be the network congestion state as seen by the receivers. This may be useful for applications where a similar consistent view is required for all the receivers, and the source is not capable of providing a multi-grade service, and hence must adapt to the receiver experiencing the worst performance. Another definition, of worst case state as seen by all receivers, is identifying the highest layer a receiver is expecting to receive in a hierarchically encoded stream. This allows the sender to adjust its transmission rate in order not to waste resources on layers that no receiver is subscribing to, and to start sending previously suppressed layers as soon as receivers subscribe to receive them. This is particularly important in the context of managing multimedia streams in collaborative sessions, because in such sessions the sender of a stream is typically simultaneously receiving multiple streams, and hence the assumption that the sender has abundant resources is not valid.

In the rest of the paper, we assume that at every instant in time each receiver is in one state  $s$ , where  $s = 1, 2, \dots, H$ .  $H$  is the highest or worst case state, and the state of a receiver may change over time.

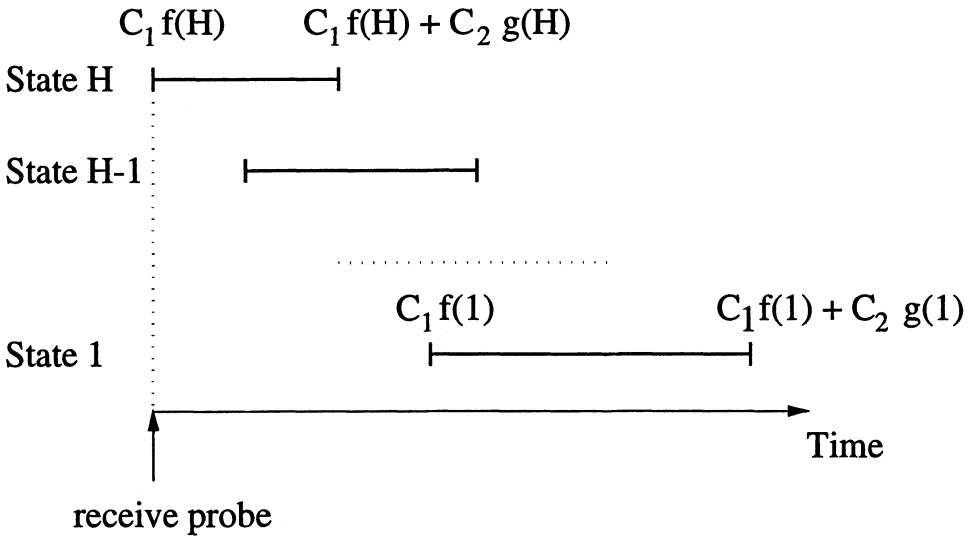
We consider the general case when neither the group size nor the round-trip time from the sender to each receiver is known. As will be shown later, this information is not necessary as the mechanism estimates the average round trip time in the group, and uses it to adjust its timeout periods.

In the proposed mechanism, the sender sends one type of probe messages, called *SolicitReply* messages, on a special multicast group which the sender and all the receivers join. The probe message contains a *RTT* field, which contains an estimate for the average round trip time from the sender to the group members. Upon receiving the *SolicitReply* probe, a receiver sets a timer to expire after a random delay period which is drawn from the interval

$$\left[ C_1 f(s) \frac{RTT}{2}, (C_1 f(s) + C_2 g(s)) \frac{RTT}{2} \right],$$

where  $f(s)$  and  $g(s)$  are two non-increasing functions of the state  $s$ ,  $C_1$  and  $C_2$  are two parameters whose values are discussed later in detail. The receiver then keeps listening to the multicast group. If the timer expires, the receiver multicasts a reply message to the whole group. The reply message contains the state information as seen by this receiver (e.g., highest layer expected to receive in a hierarchically encoded stream). On the other hand, if the receiver receives another receiver's reply before its timer expires and that reply





**Figure 2** Distribution of timeout periods according to receiver state

contains either the same or higher (worse) state, then the receiver cancels its timer and suppresses its own reply. This implies the need for careful selection of  $f(s)$ ,  $g(s)$ ,  $C_1$ , and  $C_2$  in order to avoid the reply implosion problem, while maintaining a low response time. In the subsequent subsections, we discuss in detail choices for  $f(s)$ ,  $g(s)$ ,  $C_1$ ,  $C_2$ , and  $RTT$ .

#### 4.1 Selecting the timeout functions

The objective of setting the timeout periods as a function of  $f(s)$ , and  $g(s)$  is to distribute the timeouts as in Figure 2. Receivers in higher states randomize their timeouts over periods that start earlier than receivers in lower states, thus allowing for higher state responses to suppress lower state responses. In addition, the lower state receivers randomize their timeouts over longer periods relative to higher state receivers. This is because as time elapses and no responses are generated this means that the distribution of receivers over states is biased and more receivers belong to the lower states. Thus it is desired to randomize these condensed replies over longer periods.

In order to meet these objectives,  $f(s)$  and  $g(s)$  must be non-increasing functions of  $s$ . Also,  $f(H)$  should equal 0 to avoid unnecessary delays in response time, while  $g(s) > 0$  must be satisfied for all values of  $s$  to allow for randomization of timeout periods. We chose to make  $f(s)$  and  $g(s)$  linear functions in  $s$  in order to avoid excessive delays in response time, where  $f(s) = H - s$ , and  $g(s) = f(s) + k = H - s + k$ .

The parameters  $C_1$  and  $C_2$  scale the functions  $f(s)$  and  $g(s)$ .  $C_1$  controls

the aggressiveness of the algorithm in eliminating replies from lower state receivers, while  $C_2$  controls the level of suppression of redundant replies from receivers in the same state. The values of these two parameters are explored in depth in the following sections. The value of  $k$  is set to 1. Selecting the value of  $k$  is not critical, since the parameter  $C_2$  scales  $g(s)$ , and the value of  $C_2$  can be tuned to optimize the performance of the mechanism given the selected value of  $k$ .

## 4.2 Exploring the parameter space

In this section, we attempt to find bounds for the ranges of operation of the parameters  $C_1$  and  $C_2$ . Obviously, low values for  $C_1$  and  $C_2$  are desired in order to reduce the response time. On the other hand, excessive reduction in the value of either of the two parameters may lead to inefficiency in terms of the number of produced replies possibly leading to a state of reply implosion.

In order to effect a shift in the start time of the timeout periods based on the state of the receiver, as in Figure 2,  $C_1 > 0$  must be satisfied for all  $s < H$ . This shift allows for the high state replies to suppress low state replies. Similarly,  $C_2 > 0$  must be satisfied for all values of  $s$ , in order to allow for randomization of timeout periods for receivers belonging to the same state, thus enabling suppression of redundant replies which carry the same state information.

To further bound the values of  $C_1$  and  $C_2$ , we analyze two extreme network topologies, namely: the chain and the star topologies. Given a certain distribution of receiver distances from the sender, the feedback mechanism exhibits worst case performance when the receivers are connected in a star topology with the sender at its center. This is because connecting those receivers in a star topology maximizes the distance between any pair of receivers, to the sum of their distances from the sender, and hence minimizes the likelihood of suppression of redundant replies. On the contrary connecting those receivers in a chain topology minimizes the distance between any pair, to the difference between their distances from the sender, and hence maximizes the likelihood of suppression of redundant replies. Therefore, for a given distribution of distances, and an arbitrary topology, the performance of the feedback mechanism lies somewhere in between the chain and the star cases.

### (a) Chain topology

In the chain topology, the sender is at one end of a linear list of nodes. The rest of the nodes in the list are receivers. Let  $r = \frac{RTT}{2}$  be a bound on the one way distance from the sender to any of the receivers or vice versa. Let the sender send a probe at time  $t$ . The farthest receiver receives the probe at time  $t + r$ . If this receiver is the only one in the highest state, and if it emits its reply as soon as it receives the probe, then all other receivers will have heard

this reply by time  $t + 2r$ . In order to suppress all replies from lower state receivers in this case,  $C_1 \geq 2$  must be satisfied.  $C_1 = 2$  makes the difference between the start time of two successive states equal to  $2r$ .

### (b) Star topology

In the star topology, the sender is connected to each receiver by a separate link. Any message sent from one receiver to another passes through the sender's node. Let all the receivers be at a distance  $r = \frac{RTT}{2}$  from the sender. Thus the distance between any two receivers is equal to  $2r$ .

Let  $G_s$  be the number of receivers in state  $s$ , and let  $T_s$  be the first timer to expire for receivers in state  $s$ . The expected value of  $T_s$  is  $(C_1 f(s) + \frac{C_2 g(s)}{G_s})r$ , since  $G_s$  timers are uniformly distributed over a period of  $C_2 g(s)r$ .

For receivers having the same state, if the first timer expires at time  $t$ , then all the timers that are set to expire in the period from  $t$  to  $t + 2r$  will not be suppressed, and all those that are set to expire after  $t + 2r$  will be suppressed. Therefore, the expected number of timers to expire is equal to 1 plus the expected number of timers to expire in a period of length  $2r$ , which is equal to  $1 + \frac{2G_s}{C_2 g(s)}$ . Looking at the case of  $s = H$ , since  $g(H) = 1$ , then setting  $C_2$  to any value less than 2 does not allow for suppression of any of the redundant replies from receivers in state  $H$ . Thus  $C_2 > 2$  must be satisfied.

In order to suppress all replies from receivers in state  $s - 1$ , we must have

$$\begin{aligned} T_s + 2r &\leq T_{s-1}, \\ (C_1 f(s) + \frac{C_2 g(s)}{G_s})r + 2r &\leq (C_1 f(s-1) + \frac{C_2 g(s-1)}{G_{s-1}})r, \\ \frac{g(s)}{G_s} - \frac{g(s-1)}{G_{s-1}} &\leq \frac{C_1 - 2}{C_2}. \end{aligned}$$

For values of  $G_s$  and  $G_{s-1}$  which are relatively larger than  $g(s)$  and  $g(s-1)$ , we get  $C_1 \geq 2$ , which is the same condition for  $C_1$  which we obtained from the chain topology. In Section 5, we explore the effect of  $C_2$  on the performance of the feedback mechanism using simulation experiments.

## 4.3 Estimating the round trip time

To compute the average round-trip time from the sender to the group of receivers, every probe sent is time-stamped by the sender. That time-stamp is reflected in the reply message together with the actual delay period that the receiver waited before replying. This allows the sender to compute the round-trip time to this receiver. The smoothed average round-trip time,  $srtt$ , and the smoothed mean sample deviation  $rttvar$  are computed from the received round-trip time samples, using the same technique applied in TCP (Jacobson

1988), as follows:

$$\begin{aligned} srtt &= \alpha srtt + (1 - \alpha) sample, & \alpha &= 7/8, \\ rttvar &= \beta rttvar + (1 - \beta) |srtt - sample|, & \beta &= 3/4. \end{aligned}$$

In TCP, the amount  $srtt + 4 rttvar$  is used in setting the retransmission timeouts in place of twice the round-trip time. As will be shown in Section 5, this amount is conservative and over estimates the average round-trip time to the group members. Instead we use only  $srtt$  as the estimate for average round-trip time. The recent value of  $srtt$  is carried in the  $RTT$  field of the next probe.

## 5 SIMULATION STUDY AND PERFORMANCE COMPARISON

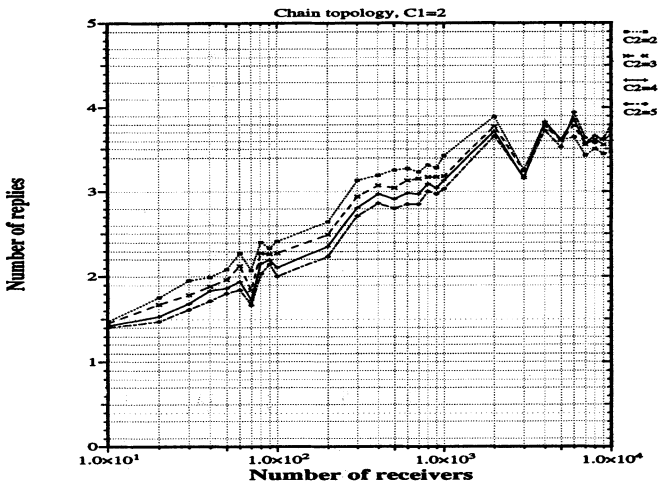
In this section, we examine various issues, related to the performance and tuning of the feedback mechanism, using simulation. First we show the ability of the new feedback mechanism to eliminate the reply implosion problem as we explore the effect of  $C_2$  on its performance. Then we examine the accuracy of the round trip time estimation algorithm. Finally, we further illustrate the scalability and robustness of the proposed feedback mechanism by contrasting it to an alternative candidate mechanism for feedback.

In order to address these issues, we ran several simulation experiments. Each experiment was setup as follows. The group size,  $G$ , and the maximum round trip time,  $RTT_{max}$ , were selected. Round trip times uniformly distributed in the interval  $[0, RTT_{max}]$  were assigned to all the receivers, except the worst case state receivers whose round trip times were uniformly distributed in the interval  $[t.RTT_{max}, RTT_{max}]$ , for investigating the effect of  $t$  over the performance, where  $0 \leq t \leq 1$ . The number of states,  $H$ , was set to 5, and each receiver was randomly assigned one of these states. The choice of 5 states (or layers) is reasonable as the state of the art hierarchical video encoders typically provide a number of layers in this range (McCanne *et al.* 1996, Senbel *et al.* 1997). Also, in applications where feedback information represents the perceived quality of service, typically 3 to 5 grades of quality are used (Bolot *et al.* 1994, Busse *et al.* 1995). The feedback mechanism was simulated under the two extreme network topologies; the chain and the star.

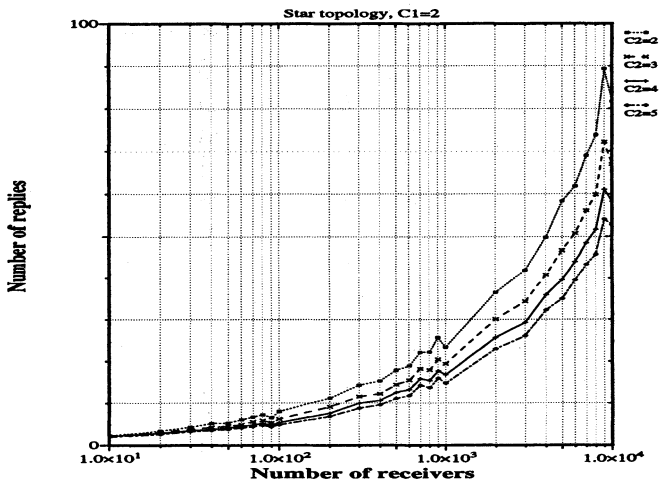
### 5.1 Bounding $C_2$

From the analysis in Section 4.2, we obtained the two conditions  $C_1 \geq 2$  and  $C_2 > 2$ . Setting  $C_1$  to its minimum value 2 eliminates replies from lower states, while avoiding unnecessary delays in response time. However, selecting an appropriate value for  $C_2$  is not as easy as such.

In Figure 3, the average number of replies is plotted for different values of



(a) chain topology



(b) star topology

Figure 3 The effect of  $C_2$  on the number of replies

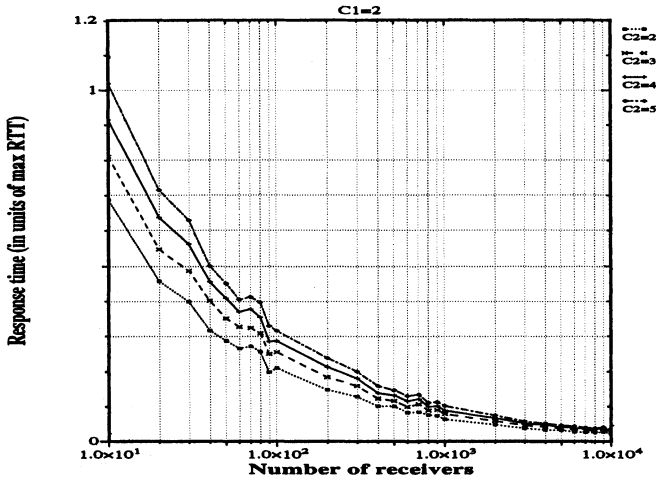


Figure 4 The effect of  $C_2$  on response time

$C_2$ . The value of  $C_1$  was set to 2, for all the experiments in this section, and the average round-trip time was used in the  $RTT$  field of the probe messages. It is clear from the figure that the performance of the feedback mechanism is not sensitive to the value of  $C_2$  in the case of the chain topology. Also, the figure shows that the reply implosion problem is totally eliminated. Moreover, over 95% of the redundant replies were correct replies (i.e., worst case state replies) which shows the robustness of the mechanism in facing network losses and its efficiency in eliminating non-worst case replies. This also means that, practically, the sender may safely react according to the first received reply. Figure 4 depicts the corresponding average response times. The response time is measured at the sender, and represents the time from sending a probe until receiving the first correct reply. The response time behavior is the same for both topologies because it is dependent on the round-trip times distribution rather than on the topology. As shown in the figure, it is bounded from above by the maximum round-trip time to the group members.

These figures suggest that  $C_2 = 4$  is a reasonable setup.  $C_2 > 4$  does not significantly reduce the number of replies, while the response time increases. As can be seen from the figures, for typical sessions with up to 100 participants (e.g., IRI sessions (Maly *et al.* 1997)), less than 10% of the receivers reply to a probe, in the worst case, while for larger sessions of thousands of participants the reply ratio is below 1.5%.

## 5.2 Evaluating the round trip time estimation technique

As mentioned in Section 4.3, the amount  $srtt + 4 rttvar$  is used in setting the retransmission timeouts in place of twice the round trip time, in TCP. Figures 5(a) and (b) compare this approach to using only  $srtt$  as the estimate for average round trip time. We chose to avoid the conservative approach of TCP, and to use only  $srtt$ , to avoid unnecessary prolonging of delay periods thus avoiding excessive delays in response time.

## 5.3 Performance comparison

Here, we further illustrate the scalability and robustness of the proposed feedback mechanism by contrasting it to an alternative candidate mechanism for feedback. The alternative mechanism uses the same approach taken by SRM (Floyd *et al.* 1995) for discriminating between receivers in setting their timeout periods based on their individual distances from the source (i.e. timeouts are selected from the interval  $[C_1 d_i, (C_1 + C_2) d_i]$  where  $d_i$  is the one way distance from receiver  $i$  to the source). This, in turn, depends on the existence of session level messages for the distance estimation process as explained in Section 3.

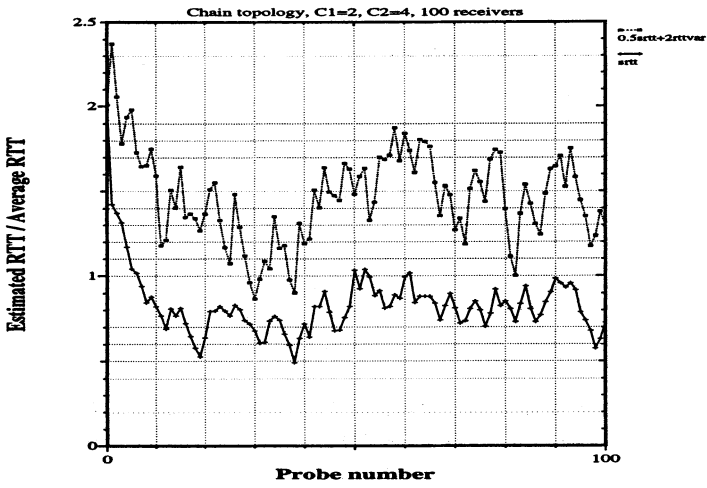
Figures 6 through 8 contrast the performance of our proposed feedback mechanism,  $A_1$ , to the alternative feedback mechanism,  $A_2$ . The comparisons are performed in two cases. In the first case, the worst case state receivers were distributed at distances in the range  $[0, RTT_{max}]$  (i.e.,  $t=0$ ). In the second case, the worst case state receivers were distributed at distances in the range  $[0.2RTT_{max}, RTT_{max}]$  (i.e.,  $t=0.2$ ).

Figure 6 shows that the total messages sent in response to a probe in the case of the new feedback mechanism,  $A_1$ , is much lower than the total response plus session messages for the alternative feedback mechanism,  $A_2$ . As discussed in Section 3, the session overhead for  $A_2$  is dependent on the session bandwidth; we depict the two cases of 1Mbps and 5Mbps sessions. For  $A_2$ , the session overhead assumed that an epoch (the time span from sending a probe until receiving the last possible reply) will take at most one second.

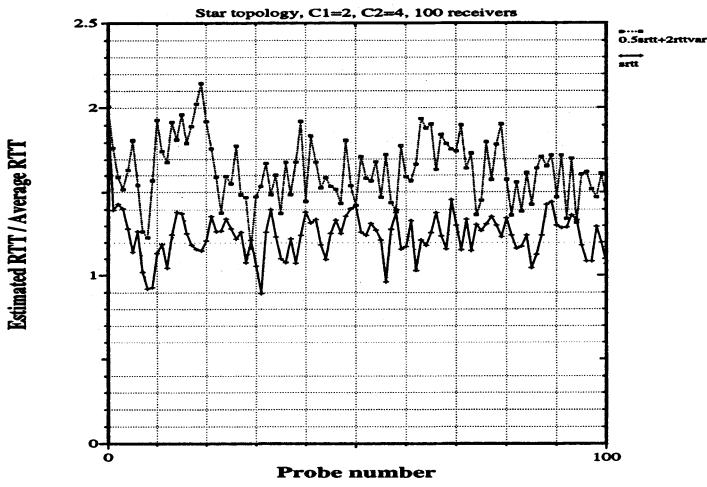
Figure 7 shows that the number of messages carrying correct worst case state information constitute almost all the total messages sent in the new algorithm  $A_1$ . In  $A_2$ , on the contrary, almost all the messages sent are overhead messages. This demonstrates the robustness of the new feedback mechanism and its tolerance to losses in reply messages.

However, Figure 8 shows that the response time of  $A_2$  is lower on the average. Nevertheless, this is not always the case for  $A_2$ , as a slight shift in the distribution of receiver distances reverses this situation and makes the response time of  $A_1$  lower. This trend continues as  $t$  increases.

From these charts, we conclude that  $A_1$  is much more robust than  $A_2$ . Also,



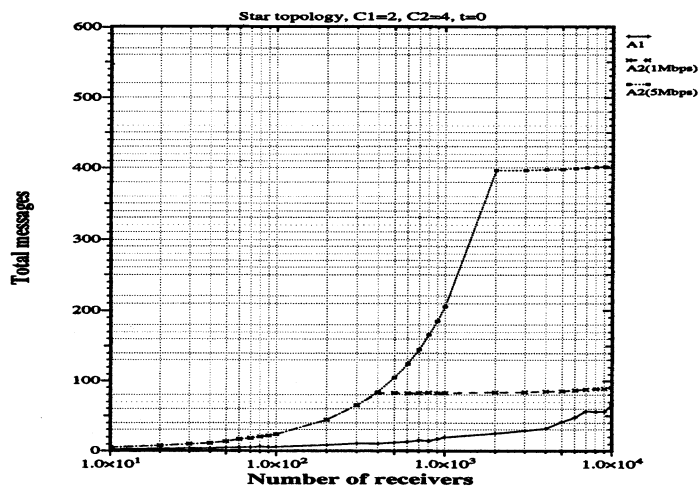
(a) chain topology



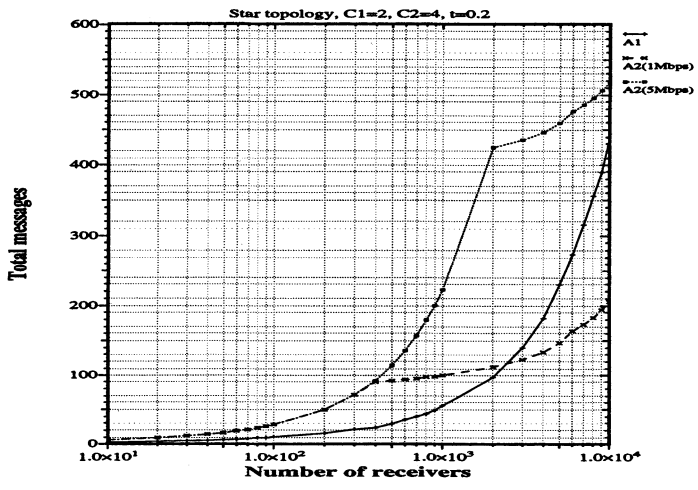
(b) star topology

Figure 5 Accuracy of RTT estimate



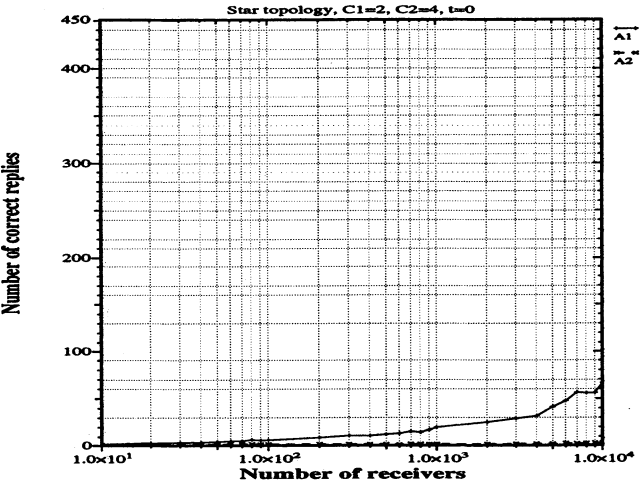


(a) t=0

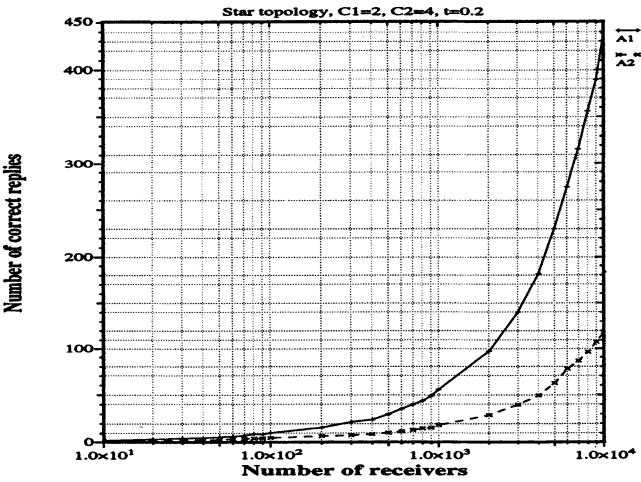


(b) t=0.2

Figure 6 Comparing total messages for the star topology

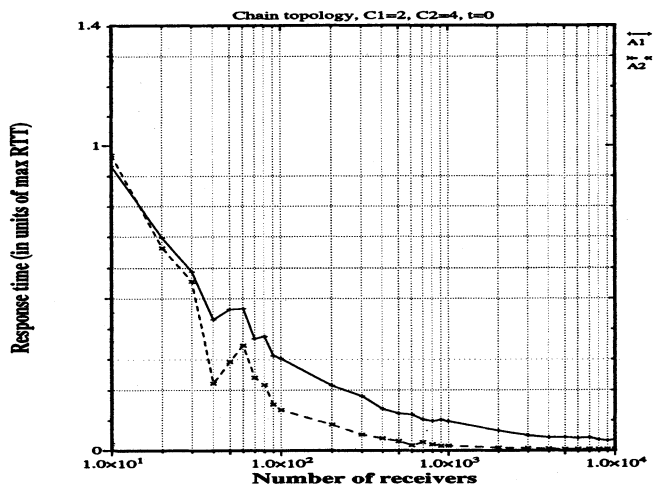


(a)  $t=0$

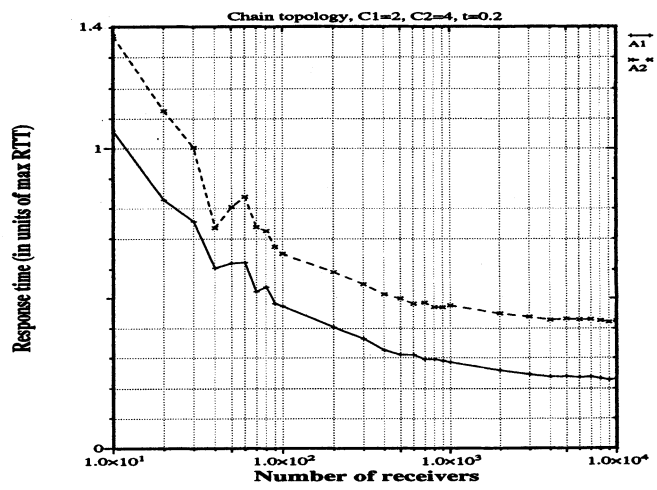


(b)  $t=0.2$

Figure 7 Comparing correct replies for the star topology



(a) t=0



(b) t=0.2

Figure 8 Comparing response time for the star topology

the total overhead of  $A_1$  is always lower than that of  $A_2$  up to sessions of few thousand participants. However, for very large sessions approaching 10000 participants, and for certain distributions of distances of receivers, the overhead of  $A_1$  starts to rise significantly. This is true for star topologies which represent worst case performance for  $A_1$ . For chain topologies, the performance of the algorithm was found to be significantly less dependent on the value of  $t$ . In the next section, we address the issue of enhancing the performance of  $A_1$  for very large sessions, and degenerate receiver distributions.

## 6 ENHANCING THE FEEDBACK MECHANISM

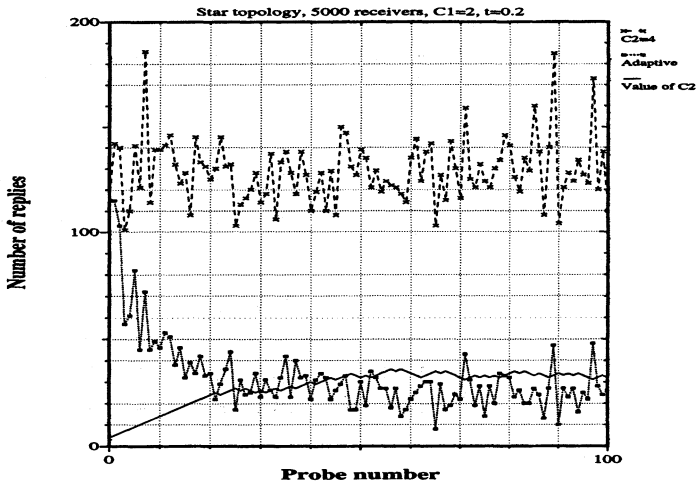
In this section, we present two enhancements for the feedback mechanism. These enhancements further improve the scalability of the feedback mechanism and reduce its overhead.

### 6.1 Adaptive feedback

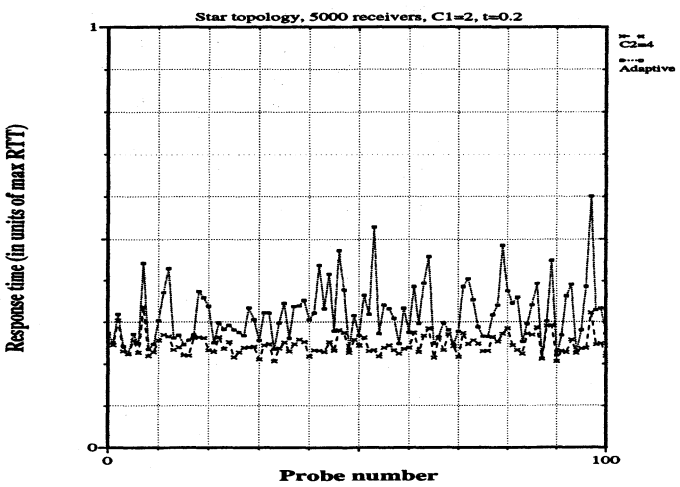
In the previous section, it was shown that the performance of the proposed feedback mechanism needs some tuning to enhance its scalability for very large groups especially in the case when the worst state receivers are far from the sender, and most importantly far from each other. We focus on the worst state receivers because the outcome of the simulation experiments, discussed in the previous section, shows that almost all the excess replies that are generated in these cases are redundant worst case replies. This means that the shift in the start time of the timeout periods is still effective in eliminating replies from lower state receivers. Thus the parameter  $C_1$  does not need tuning. It is the parameter  $C_2$  which needs to be adapted to support very large groups. In other words, as the group size increases too much, the fixed value of  $C_2 = 4$  no longer suffices to effectively suppress enough redundant replies. To this end we developed a simple adaptive algorithm that the sender uses to adapt the value of  $C_2$  dynamically based on the number of received redundant replies. The sender counts the number of redundant worst state replies in response to a probe in the variable *dups*. Note that based on our previous results, the sender can safely count all replies coming in response to a probe assuming they are all worst state replies. Before sending a probe, the sender computes a new value for  $C_2$  and appends it to the probe message. This value is used by the receivers in computing their random timeout periods. The algorithm which the sender applies is as follows.

```

AvgDups =  $\alpha$  AvgDups + (1- $\alpha$ ) dups;
If AvgDups > THRESHOLD
     $C_2 = \text{Min}(C_2+1, \text{MAX\_}C_2)$ ;
Else
```



(a) number of replies



(b) response time

Figure 9 Effect of adaptive feedback

$$C_2 = \text{Max}(C_2-1, \text{MIN\_}C_2);$$

Figures 9(a) and (b) compare the performance of the static and adaptive feedback. In this simulation experiment,  $\text{MIN\_}C_2$ ,  $\text{MAX\_}C_2$ ,  $\text{THRESHOLD}$ , and  $\alpha$  were set to 4, 50, 25, and 0 respectively. The figures show the ability of the simple adaptive algorithm to reduce the number of redundant replies drastically, without significant delay in response time. The tradeoff, however, is that it takes the sender a longer time before it can declare that the current epoch is over and no further replies will be received. Typically, the sender sends a new probe only at the end of an epoch, to avoid overlapping replies. The sender can always safely terminate an epoch after an amount of time equal to  $(C_1 f(h) + C_2 g(h) + 2) \frac{RTT}{2}$  from sending a probe, where  $h$  is the highest state received in a reply to the current probe. After sending a probe, the sender sets a timer to expire after  $RTT$  plus the longest possible timeout period in the lowest state, for ending the epoch. As it receives replies, it adjusts this timer according to the above equation which is linearly proportional to  $C_2$ .

A more aggressive approach for ending an epoch without relying on  $C_2$  would be to terminate the epoch after a period of time equal to  $RTT$  from the time of receiving the first reply. This aggressive approach safely assumes that any reply is coming from the highest state in the group. It attempts to give enough time for this reply to propagate to all other receivers and cause them to suppress their replies, if they haven't already sent it. The approach relies on the heuristic assumption that  $RTT \cong \frac{RTT_{\text{max}}}{2}$ .

If it is desired to limit the bandwidth taken by the reply packets to  $R$ , then the  $\text{THRESHOLD}$  value can be set as a function of  $R$ . A simple approach is to set  $\text{THRESHOLD} = \frac{R}{\text{Reply size}} \times \text{Epoch duration}$ .

## 6.2 Passive feedback

The feedback mechanism, as described, keeps polling the receivers all the time. As soon as the sender determines that an epoch has ended, it immediately sends the next probe. While these probes are important for synchronizing the operation of the mechanism and avoiding potential spontaneous chains of status change notifications from receivers, yet in situations where the states of the receivers are stable for relatively long periods of time, this repeated probing is unnecessary.

One possible solution to optimize the performance of the feedback mechanism in such cases is to make the sender exploit the flexibility in spacing the probes, by increasing the idle time between ending an epoch and sending the following probe. However, this approach negatively affects the responsiveness of the feedback mechanism, especially when a change in state occurs after a relatively long stable state.

Another solution is to switch the feedback mechanism into *passive* mode whenever these relatively long stable states occur. When the sender gets similar state feedback from  $n$  consecutive probes, it sends a probe with a *passive flag* set, and carrying the current highest state  $h$ . Receivers do not respond to this probe, and the sender enters a passive non-probing mode. If a receiver detects that its state has risen above  $h$ , it immediately sets a timer in the usual way to report its state. On receiving a reported new higher state, each receiver updates the value of  $h$ . Similarly, if a highest state receiver detects that its state has fallen below  $h$ , it sets a timer in the usual way. However, when the receivers hear a report below  $h$  they do not update the value of  $h$  (as other receivers may be still in the  $h$  state). The sender, on receiving this report, switches back to the active probing mode, and the same cycle repeats.

## 7 CONCLUSION

In this paper, we presented a scalable and robust feedback mechanism for supporting adaptive multimedia multicast systems. Providing the source of a stream with feedback information about the used layers of the stream is crucial for the efficient utilization of the available resources. The feedback mechanism allows the sender to always send only layers for which interested receivers exist, and to suppress unused layers.

Simulation results showed that the proposed feedback mechanism scales well for groups of up to thousands of participants. For typical sessions with up to 100 participants (e.g., IRI sessions (Maly *et al.* 1997)), less than 10% of the receivers reply to a probe, in the worst case, while for larger sessions, of a few thousands of participants, the reply ratio is below 1.5%. The response time was found to be always below the maximum round-trip time from the sender to any of the group members.

The mechanism was shown to be robust in facing network losses, and to be more efficient than mechanisms which rely on session level messages for estimating individual round-trip times from each receiver to the sender. In addition, adaptive enhancements for supporting groups of up to 10,000 participants were proposed and shown to be effective in reducing the number of replies without a significant effect on response time.

Currently, we are incorporating the feedback mechanism in the *Quality of Session* control platform described in (Youssef *et al.* 1998), for further exploration and experimentation.

## REFERENCES

- Bolla, R., Marchese, M. and Zappatore, S.(1997) A Congestion Control Scheme for Multimedia Traffic in Packet Switching Best-Effort Networks. *Proceedings of the Second European Conference on Multime-*

- dia Applications, Services and Techniques (ECMAST'97)*, Milan, Italy, May 1997.
- Bolot, J., Turetti, T. and Wakeman, I.(1994) Scalable Feedback Control for Multicast Video Distribution in the Internet. *ACM SIGCOMM*, October 1994.
- Bolot, J. and Turetti, T.(1994) A Rate Control Mechanism for Packet Video in the Internet. *Proceedings of IEEE INFOCOM'94*.
- Busse, I., Deffner, B. and Schulzrinne, H.(1995) Dynamic QoS Control of Multimedia Applications based on RTP. *Second Workshop on Protocols for Multimedia Systems*, Salzburg, Austria, October 1995.
- Cheung, S., Ammar, M. and Li, X.(1996) On the Use of Destination Set Grouping to Improve Fairness in Multicast Video Distribution. *Proceedings of IEEE INFOCOM'96*, San Francisco, CA, March 1996.
- Deering, S. and Cheriton, D.(1990) Multicast Routing in Internetworks and Extended LANs. *ACM Trans. on Computer Systems*, 8(2), 85-110, May 1990.
- Floyd, S., Jacobson, V., Liu, C., McCanne, S. and Zhang, L.(1995) A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing. *ACM SIGCOMM*, August 1995.
- Gupta, A., Howe, W., Moran, M. and Nguyen, Q.(1995) Resource Sharing for Multi-Party Real-Time Communication. *Proceedings of IEEE INFOCOM'95*.
- Jacobson, V.(1988) Congestion Avoidance and Control. *ACM Computer Communication Review*, 18(4), 314-329, August 1988.
- Li, X. and Ammar, M.(1996) Bandwidth Control for Replicated-Stream Multicast Video Distribution. *Proceedings of the Fifth IEEE Symposium on High Performance Distributed Computing (HPDC-5)*, Syracuse, NY, August 1996.
- Maly, K., Abdel-Wahab, H., Overstreet, C.M., Wild, C., Gupta, A., Youssef, A., Stoica, E. and Al-Shaer, E.(1997) Interactive Distance Learning over Intranets. *IEEE Internet Computing*, 1(1), 60-71, January 1997.
- McCanne, S. and Jacobson, V.(1996) Receiver-driven Layered Multicast. *ACM SIGCOMM*, Stanford, CA, August 1996.
- McCanne, S., Vetterli, M. and Jacobson, V.(1997) Low-complexity Video Coding for Receiver-driven Layered Multicast. *IEEE Journal on Selected Areas in Communications*, 16(6), 983-1001, August 1997.
- Senbel, S. and Abdel-Wahab, H.(1997) A Quadtree-based Image Encoding Scheme for Real-Time Communication. *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, Ottawa, Canada, June 1997.
- Schulzrinne, H., Casner, S., Fredrick, R. and Jacobson, V.(1996) RTP: A Transport Protocol for Real-Time Applications. *RFC 1889*, January 1996.
- Willebeek-LeMair, M. and Shae, Z.(1997) Videoconferencing over Packet-



- Based Networks. *IEEE Journal on Selected Areas in Communication*, **15(6)**, 1101-1114, August 1997.
- Youssef, A., Abdel-Wahab, H. and Maly, K.(1998) The Software Architecture of a Distributed Quality of Session Control Layer. *Proceedings of the Seventh IEEE Symposium on High Performance Distributed Computing (HPDC-7)*, Chicago, IL, July 1998.
- Youssef, A., Abdel-Wahab, H., Maly, K. and Gouda, M.(1997) Inter-Stream Adaptation for Collaborative Multimedia Applications. *Proceedings of the Second IEEE Symposium on Computers and Communications (ISCC'97)*, Alexandria, Egypt, July 1997.
- Zhang, L, Deering, S., Estrin, D., Shenker, S. and Zappala, D.(1993) RSVP: A New Resource ReSerVation Protocol. *IEEE Network Magazine*, September 1993.

## 8 BIOGRAPHY

Alaa Youssef is a PhD candidate and research assistant in computer science at Old Dominion University. His research interests include networking support for multimedia systems, resource management in heterogeneous distributed systems, and multimedia collaborative and tele-teaching systems. Youssef received his BSc and MSc in computer science from Alexandria University, Egypt, in 1991 and 1994, respectively.

Hussein Abdel-Wahab is a professor of computer science at Old Dominion University, an adjunct professor of computer science at the University of North Carolina at Chapel Hill, and a faculty member at the Information Technology Lab of the National Institute of Standards and Technology. His main research interests are collaborative desktop multimedia conferencing systems and real-time distributed information sharing. Abdel-Wahab received a PhD in computer communications from the University of Waterloo. He is a senior member of IEEE Computer Society and a member of the ACM.

Kurt Maly is a Kaufman professor and chair of computer science at Old Dominion University. His research interests include modeling and simulation, very high performance network protocols, reliability, interactive multimedia remote instruction, Internet resource access, and software maintenance. Maly received a PhD in computer science from the Courant Institute of Mathematical Sciences, New York University. He is a member of the IEEE Computer Society and the ACM.

# A Scalable Protocol For Reporting Periodically Using Multicast IP

*Ljubica Blazević, Eric Gauthier*  
*ICA, Swiss Federal Institute of Technology Lausanne*

## Abstract

We propose a protocol that controls the members of a multicast group that send periodically status reports to all members. The protocol, called Multicast Access Protocol (MAP), limits the number of concurrent multicast reports as the group size becomes large. MAP is a decentralised protocol that provides an access control mechanism to an IP multicast group. The protocol supports members joining and leaving the group dynamically as well as changes in the underlying network topology. MAP is a self-configuring mechanism and requires every member to keep only local information independent of the group size without using random timers. We describe the protocol both formally and informally.

## Keywords

multicast IP, access control, distributed protocol, self-configuration

## 1 INTRODUCTION

We address the problem of limiting the number of concurrent members of a multicast group that send periodically status information to all other members. This problem arises in the Scalable Reliable Multicast protocol where each member must report the largest sequence number of the data packet that it has received from each sender to the whole group [1]. The periodical sending of reports does not scale to large groups since the number of reports received by a member is proportional to the number of group members. Multicast IP, as defined in [2], does not control the number of members in a given multicast group that can send data simultaneously. One solution to reduce the number of received reports is for every member to send its report to a server that combines the received reports and send a combined report to all members. However the server becomes a bottleneck as the group size increases. To overcome this limitation Mark Handley [3] proposed to use multiple servers: each server receives the reports from a different subgroup of members and sends a combined report to its members and to the other servers. This scheme does not support well new members joining and leaving the multicast group as

well as changes in the underlying topology since the servers must be statically configured. An adaptive alternative is to use a two-level hierarchy [4] in the following way: each member reports to a local representative member elected using random timers which then sends a combined report to all other local representatives. Each representative sends then a combined report to all its local members. However in the absence of hierarchy this approach requires that each member maintains a table of distance to all other members in order to adjust the random timers. Using a two level hierarchy a member must only keep a table of distance to all other members having the same representative. Also in this case each representative is also required to maintain a table of distance to all other representatives.

We propose in this paper a different approach called Multicast Access Protocol (MAP) that does not require random timers. A member that uses MAP does not keep a distance table to all members but only to a subset of the members. Let  $d(i, j)$  be the number of hops that a multicast packet sent by member  $i$  goes through before arriving at member  $j$ . We assume that  $d(i, j) = d(j, i)$ . We say that member  $i$  is a *neighbor* of member  $j$  if  $d(i, j) < d(k, j)$  for all members  $k$  such that  $d(i, k) < d(i, j)$ . Thus a member keeps a table of distances only to its neighbors. In the worst case a member keeps a table of distance to all other members. This worst case occurs only if  $d(i, j) = d(i, k)$  for all members  $i, j, k$ . In general the number of members in the table of distance should be independent of the group size. MAP can also be used in a two-level hierarchy but in this paper we restrict the description of MAP for the case of a flat hierarchy.

MAP relies on the notion of a *grant* as defined in the SMART[5] protocol originally designed for ATM. SMART guarantees that, at a given instant, only one end-system is concurrently sending data on a given connection of an ATM multicast tree. In a similar way, MAP limits the number of concurrent report senders to a maximum number  $n$  which is fixed initially. The idea of applying SMART to IP was suggested to the authors by Jean-Yves Le Boudec [6].

To achieve this on multicast IP, MAP builds and maintains  $n$  spanning trees of the multicast group members, where  $n$  is the maximum number of reports that can be sent at the same time. All spanning trees of a given group are the same except that they can be directed differently. Each spanning tree is rooted and directed inward so that a member needs to keep track of one of its neighbor, called its *parent*. The member of a spanning tree without parent is called *root*. If a member of a spanning tree is a *root* and it is allowed to send a report then we say the member is a *leader*. A member and its neighbors exchange control informations by sending MAP control packets by unicast. Members retransmits state information if the protocol requires it and do not require a reliable transport protocol.

The purpose of a MAP control packet is to change the root of a particular spanning tree and to ensure that every member becomes root of this tree in its turn. When a member becomes root of a spanning tree the protocol

determines if it is the leader of this tree and is thus allowed to send one report. Assuming that no member join or leave the group from time  $t = 0$ , if  $n(i, t)$  is the number of times member  $i$  sends a report during  $[0, t]$ , then MAP ensures that  $|n(i, t) - n(j, t)| \leq 1$  for all members  $i, j$ , and time  $t$ .

MAP constructs the spanning trees using the reports sent by the members. Every host is always free to join or leave the multicast group and the spanning trees are reconfigured accordingly. MAP ensures that adjacent members on the spanning trees are neighbors. The spanning trees are completely distributed and do not rely on any centralised coordinator. The spanning tree is built dynamically as members join and leave the multicast group.

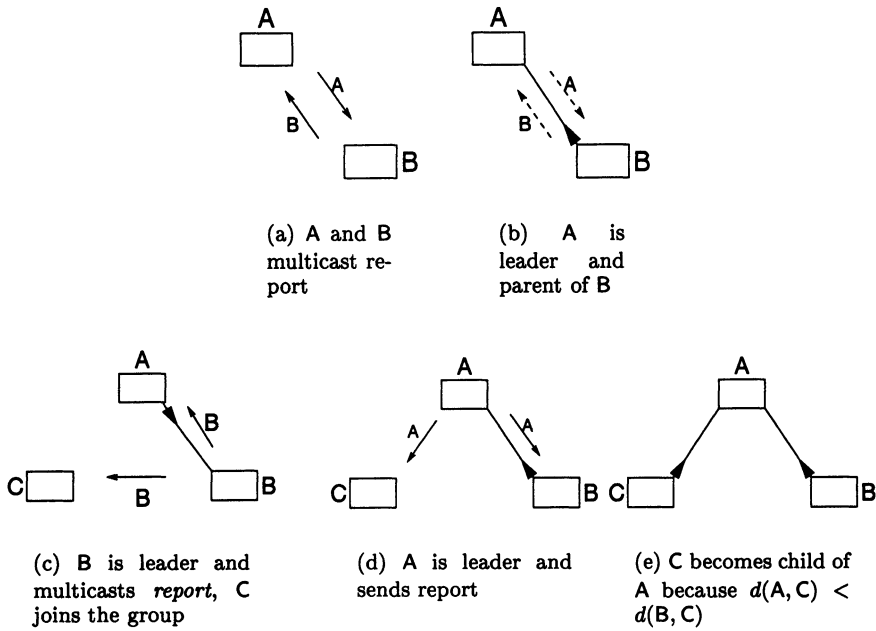
The protocol requires that every member keeps the following state information 4bits per spanning tree and 3 bits per spanning tree and per neighbor. MAP uses only one safety timer that guarantees the spanning trees are loop-frees. A MAP control packet carries  $2 + \lceil \log n \rceil$  bits where  $n$  is the maximum number of reports that can be sent at the same time. In the two following sections we assume that  $n = 1$ . The paper is organised as follows: Section 2 gives an overview of the protocol for the case where it allows only one member to send a report concurrently. Then the protocol in the case it allows only one member to send a report concurrently is formally specified in Section 3. Section 4 lists some open issues.

## 2 OVERVIEW OF THE PROTOCOL

This section describes the general mechanisms of the MAP protocol for the case where only one report can be sent at the same time. A detailed specification of the protocol is given in Section 3.

The construction of a spanning tree is shown in Figure 1. Hosts A and B join simultaneously a multicast group identified by address mcast. A and B start their timer  $T_1$ . When their timer  $T_1$  expires A and B are both leaders and multicast a report, see Figure 1(a). A solid arrow shows a report in transit whereas a dashed arrow indicates that the report was received. B receives the report of A and determines that A is its parent since B's IP address is smaller than A's IP address. The parent-child relationship is shown by a filled arrow pointing out of B towards A. A is the leader of the spanning tree A-B, see Figure 1(b). B sends a MAP control packet by unicast to A. Meanwhile host C joins the multicast group and starts its timer  $T_1$ . A becomes the child of B and sends a MAP control packet to B. B receives the packet from A and becomes leader and multicasts a report, see Figure 1(c). Immediately B becomes the child of A and sends a MAP control packet to A. C receives the report of B. As soon as A receives the MAP packet from B, A becomes leader and multicasts a report, see Figure 1(d). C receives the report of A.

Timer  $T_1$  at C expires and C becomes child of A (Figure 1(e)) because  $d(A, C) < d(B, C)$ .



**Figure 1** Construction of Tree A-B-C

As more hosts join the multicast group, the spanning tree is dynamically updated as in the previous example.

A MAP control packet contains a 2 bit sequence number.

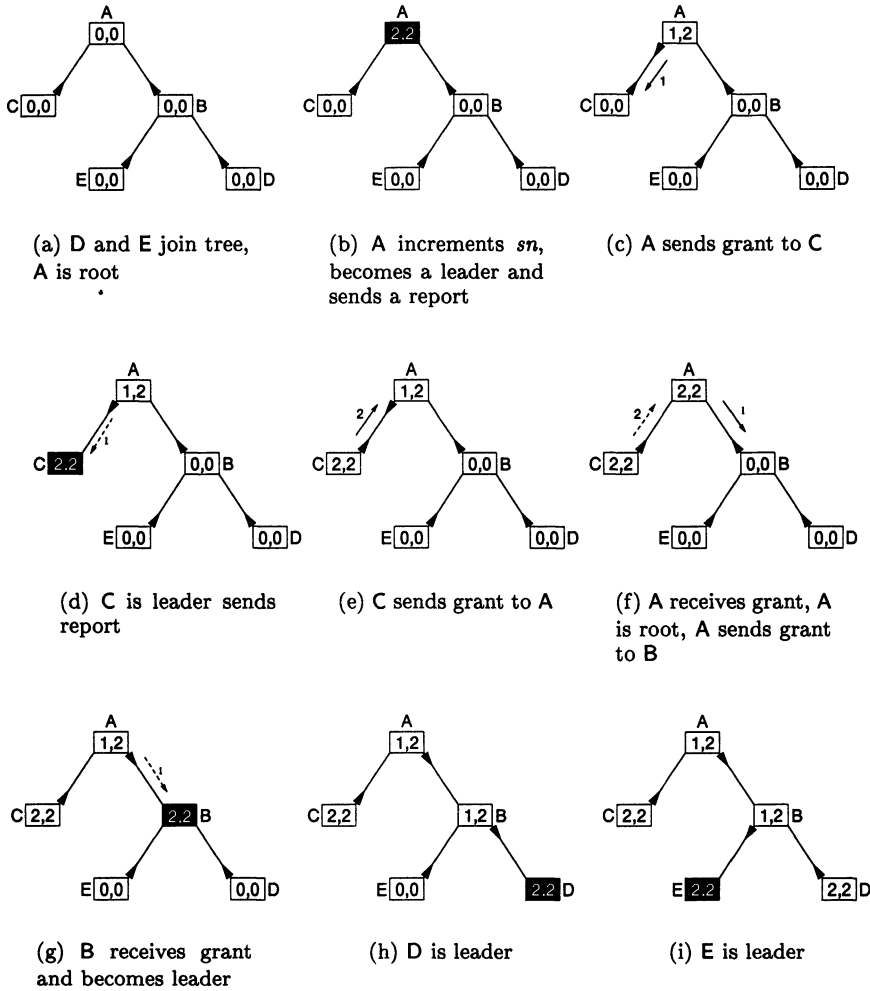
A member keeps two state variables:

*sn* a sequence number that takes the values 0,1 and 2, and that take initially value 0.

*ln* the last value of *sn* at which the member was a leader.

The access control to the multicast tree is shown in more details in Figure 2.

The spanning tree consists of five hosts A, B, C, D and E. A is parent of B and C and B is parent of E and D. A is root of the tree A-B-C-D-E. Initially *sn* and *ln* take the value 0, see Figure 2(a). A increments *sn* modulo 3 since no neighbor has its *sn* equal to 1 and 2, see Figure 2(b). A is now the leader since  $ln = (sn + 1) \bmod 3$ , sends a report and sets  $ln := sn$ , see Section 3. The state values of A are showed in white on a black background as long as A is a leader. A sends a control packet with *sn* to C, see Figure 2(c). This control packet is a grant [5]. C receives the grant from A, becomes the root, sends a control packet to A to acknowledge the reception of the grant that carries the same sequence number as the grant. C increments its *sn*, becomes a leader and sends a report, see Figure 2(d). C sends a grant to A since it has no other



**Figure 2** Single Access Control on Tree A-B-C-D-E

neighbor, see Figure 2(e). A receives the grant, becomes the root and sends an acknowledgement to C. A sends a grant to B since B has a sequence number equal to  $(2+1) \bmod 3$ , see Figure 2(f). B receives grant, becomes a root, sends an acknowledgement to A, increments its  $sn$  twice modulo 3 and becomes a leader, see Figure 2(g). Similarly D and E become leader and send a report, see Figure 2(h) and Figure 2(i). This example shows how the protocol ensures that every member sends a report in turn.

### 3 SPECIFICATION OF THE PROTOCOL

This section presents a formal specification of MAP for the case where only one report can be sent at a time.

We recall from Section 2 that a member keeps two state variables:

- $sn$  a sequence number that takes the values 0,1 and 2, and that take initially value 0.
- $ln$  the last value of  $sn$  at which the member was a leader.

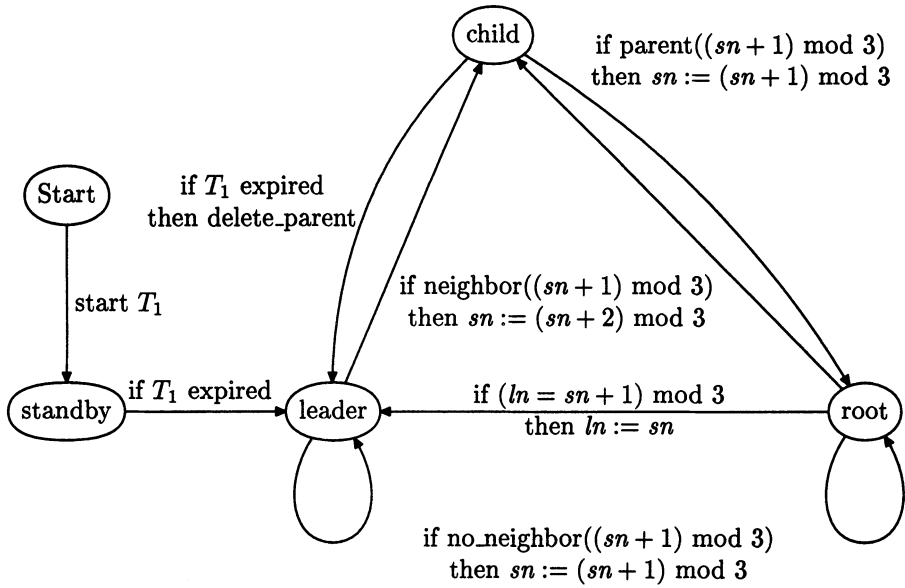
A member keeps also a safety timer  $T_1$  which is set to value that represents the maximum delay tolerated by a member to send a report. A second timer  $T_2$  is used to resend a control packet to a parent if the grant or its acknowledgement were lost. In addition a member keeps a distance table which consists per neighbor of:

- the IP address of the neighbor.
- $d(\text{member}, \text{neighbor})$  in number of hops.
- a parent flag (one bit) that is set if neighbor is parent and else is reset.
- a 2 bit sequence number received in the last MAP control packet from this neighbor.

The four following functions are associated with the distance table.

- $\text{neighbor}(x)$  searches the table and sets the parent flag for the first neighbor it finds with a sequence number equal to  $x$  and returns true, else it returns false.
- $\text{no\_neighbor}(x)$  searches the table and returns true if no sequence number equal to  $x$  was found, else it returns false.
- $\text{reset\_parent}(x)$  searches the table, if there is a neighbor with parent flag set and sequence number equal to  $x$ , returns true and finds the neighbor with the parent flag set and  $d(\text{member}, \text{neighbor})$  is minimum, resets this neighbor flag and delete all other neighbors with parent flag set, else returns false.
- $\text{delete\_parent}$  searches the table for a neighbor with the parent flag set and deletes the neighbor from the table.

Each member that implements MAP is modelled as a finite state machine. A transition has a label of the form: **if** (condition) **then** actions. A transition is fireable if the condition is true. Fireable transitions are executed one at a time. When a transition is executed all its actions are executed as one atomic action. If several transitions are fireable at the same time one is chosen at random. The state machine is shown in Figure 3.



**Figure 3** The state machine per spanning tree

**State standby.** When a new member joins the multicast group, it starts timer  $T_1$  and enters state standby. If a member receives a report while in state standby and its distance table is empty it inserts the sender as a new neighbor with sequence number equal to 0 and a parent flag reset. If the member receives a report and the distance table has one entry, and  $d(\text{member}, \text{sender}) < d(\text{member}, \text{neighbor})$ , the neighbor is deleted and the sender is inserted as a new neighbor with sequence number equal to 0 and a parent flag reset, else the distance table remains the same. When state is standby and timer  $T_1$  expires, the state changes to leader and only if the distance table is empty the member multicasts a report.

**State leader.** While in state leader if there is no neighbor in the table with sequence number equal to  $(sn+1) \bmod 3$ ,  $sn$  is incremented modulo 3. If the state is leader and there is a neighbor in the table with sequence number equal to  $(sn+1) \bmod 3$ ,  $sn$  is incremented twice modulo 3, timer  $T_1$  is started and send a control packet to neighbor with  $sn$  and the state changes to child.

**State child.** If state is child and  $T_1$  expires, the parent neighbor is deleted from the table and the state changes to standby. While in state child if a report is received and  $d(\text{member}, \text{sender}) < d(\text{member}, \text{parent})$ , the sender is added as a new neighbor in the table with sequence number equal to 0 and a parent flag set. If in state child and there is a neighbor with parent flag set and sequence number equal to  $(sn+1) \bmod 3$  then send a packet to the neighbor with sequence number  $(sn+1) \bmod 3$  and starts timer  $T_2$ . Member



finds the neighbor with the parent flag set and  $d(\text{member}, \text{neighbor})$  minimum, resets this neighbor flag, delete all other neighbors with parent flag set,  $sn$  is incremented modulo 3, the state changes to root. If in state child and timer  $T_2$  expires resend a packet with  $sn$  to parent. If in state child and receive packet from parent with sequence number equal to  $sn$ , stop  $T_2$ .

*State root.* While in state root if there is no neighbor in the table with sequence number equal to  $(sn + 1) \bmod 3$ ,  $sn$  is incremented modulo 3. If the state is root and there is a neighbor in the table with sequence number equal to  $(sn + 1) \bmod 3$ ,  $sn$  is incremented twice modulo 3, send a control packet to neighbor with  $sn$  and the state changes to child. If the state is root and  $ln = (sn + 1) \bmod 3$ ,  $ln := sn$  and the state changes to leader.

## 4 OPEN ISSUES

MAP presents an alternative to the session message mechanism in SRM [4]. The two approaches should be compared in a flat hierarchy scenario as well as for a two-level hierarchy. It is not clear also which of the two alternatives will be more robust to packet losses. Also many interesting questions remains to be answered about MAP such as the overhead for all members to send one report each as well as the associated latency.

## 5 CONCLUSIONS

We have proposed in this paper a protocol that controls the members of a multicast group that send periodically status reports to all members. The protocol, called Multicast Access Protocol (MAP), limits the number of concurrent multicast reports as the group size becomes large. MAP is a decentralised protocol that provides an access control mechanism to an IP multicast group. The protocol supports members joining and leaving the group dynamically as well as changes in the underlying network topology. MAP is a self-configuring mechanism and requires every member to keep only local information independent of the group size without using random timers. Assuming that no member join or leave the group from time  $t = 0$ , if  $n(i, t)$  is the number of times member  $i$  sends a report during  $[0, t]$ , then MAP ensures that  $|n(i, t) - n(j, t)| \leq 1$  for all members  $i, j$ , and time  $t$ . MAP was showed to be free of deadlocks by extensive simulations using the model checker SPIN [7],[8]. including in case of message loss, duplication and reordering. An implementation was tested successfully in a LAN environment. A complete proof of correctness of a protocol is left for further study.

## REFERENCES

- [1] S. Floyd, V. Jacobson, S. McCanne, C.-G. Liv, and L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing," in *Proc. of ACM SIGCOMM'95*, USA, August 1995, pp. 342–356.
- [2] S. Deering, "Host Extensions for IP Multicasting," Internet RFC1112, Aug. 1989.
- [3] M. Handley, "SAP: Session Announcement Protocol," Internet Draft, 1996.
- [4] P. Sharma, D. Estrin, and S. Floyd, "Scalable Session Messages in SRM using Self-configuration," submitted to SIGCOMM, 1998.
- [5] E. Gauthier, J.-Y. Le Boudec, and Ph. Oechslin, "A many-to-many multicast protocol for atm," *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 458–472, April 1997.
- [6] J.-Y. Le Boudec, "private communication," .
- [7] G. J. Holzmann, *Design and Validation of Computer Protocols*, Prentice Hall, 1991.
- [8] G. J. Holzmann, "The Model Checker SPIN," <http://netlib.att.com/netlib/att/cs/home/holzmann-spin.html>.

## 6 BIOGRAPHY

*Ljubica Blazević* received the B.S. degree in 1993 from the Faculty of Electrical Engineering, Belgrade, Yugoslavia. From 1993 to 1996 she worked as a R&D engineer at Institute "Mihajlo Pupin" in Belgrade. She is currently a Research Assistant at ICA, Swiss Federal Institute of Technology Lausanne, where she is engaged in research on multicast routing and traffic control.

*Eric Gauthier* is born in Zurich, Switzerland, in 1970. He received the B.Eng. degree in electrical engineering from McGill University, Montreal, Canada, in 1992, and the Master degree from INRS-Télécommunications, Verdun, Canada, in 1995. He is currently working toward the Ph.D. degree at the Swiss Federal Institute of Technology Lausanne.

# A Scalability Scheme for the Real-time Control Protocol

*R. El-Marakby, D. Hutchison*

*Computing Dept., Lancaster University, Lancaster LA1 4YR, U.K.*

*Tel.: (+44)-1524-65201 Ext. 94538, Fax: (+44)-1524-593608*

*E-mail: <randa,dh>@comp.lancs.ac.uk*

## **Abstract**

Recently, some problems related to using the Real-time Control Protocol (RTCP) in very large dynamic groups have arisen. Some of these problems are: feedback delay, increasing storage state at every member, and ineffective RTCP bandwidth usage, especially for receivers that obtain incoming RTCP reports through low bandwidth links. In addition, the functionality of some fields (e.g. packet loss fraction) in the Receiver Reports (RRs) becomes questionable as, currently, an increasing number of real-time adaptive applications are using receiver-based rate adaptive schemes instead of rate adaptation schemes based on the sender.

This paper presents the design of a scalable RTCP (S-RTCP) scheme. S-RTCP is based on a hierarchical structure in which members are grouped into local regions. For every region, there is an Aggregator (AG) which receives the RRs sent by its local members. The AG extracts and summarises important information in the RRs, derives some statistics, and sends them to a Manager. The Manager performs additional statistical analysis to monitor the transmission quality and to estimate regions which are suffering massively from congestion.

We believe that our S-RTCP alleviates some of the RTCP scalability problems encountered in very large dynamic groups and makes effective use of RRs with regard to the current changing requirements of real-time adaptive applications in the Internet today.

## **Keywords**

RTCP scalability, RR, TTL, AG, LAG, AGR, Manager

## 1 INTRODUCTION

Today, the Real-time Transport Protocol (RTP) is widely deployed in most MBone applications over the Internet involving multiple senders and receivers. The Real-time Control Protocol (RTCP) which is RTP's control protocol is used mainly in adaptive applications where the sender changes its rate of data transmission in order to suit the current state of the network.

Some problems have arisen when RTCP has been used in very large dynamic multicast groups (Rosenberg, 1997a), (Rosenberg, 1997b), (Schulzrinne, 1997). Firstly, because the RTCP reporting interval grows linearly with the group size, a feedback delay occurs. Consequently, infrequent feedback reports are sent and timely reporting does not occur. Second, each member has to keep track of every other member in the group, thus a storage scalability problem can appear. Third, a flood of initial RTCP reports multicast to the whole group can occur when large number of members join at the same time. As a result, members can be flooded with these reports, especially the ones connected to the network through low bandwidth links, and the network may be congested. This problem occurs (Aboba, 1996) if the reporting members are not implementing the reconsideration algorithm described in (Rosenberg, 1997a). Fourth, for receivers connected through low bandwidth links, the RTCP bandwidth available could be used more effectively than is presently the case.

Today, in the Internet, some of the requirements for real-time adaptive applications are changing. An increasing number of the current multicast applications prefer to use receiver-based rate adaptation schemes instead of sender-based rate adaptation schemes to adapt to congestion in the network. In sender-based rate adaptations, when congestion occurs, the sender decreases its rate of data transmission to suit the receiver with the lowest capabilities. Receiver-based adaptive applications have the advantage of accommodating to the heterogeneous capabilities and conflicting bandwidth requirements of different receivers in the same multicast group (McCanne, 1996). Also, adaptation is done immediately instead of waiting for the sender to adapt. With the appearance of these kinds of receiver-based applications, we ask the question: what is the function of the RTCP Receiver Reports (RRs) and how can RRs be used effectively in the current Internet?

We designed a hierarchical scheme which groups members in local regions. Members in each local region send their RRs locally to an Aggregator (AG) in the same region. The AG summarises important information in the RRs, derives some statistics, then sends this information to a Manager. The Manager does some monitoring and diagnosis functions to estimate which regions are suffering highly from congestion and to evaluate the quality of the transmitted data.

This paper proceeds as follows. In Section 2, some background information about RTP/RTCP functionality is presented. Section 3 presents some of the scalability problems of RTCP feedback reports. In Section 4, we describe our Scalable RTCP (S-RTCP) scheme. Section 5 presents some of the benefits of using S-RTCP. Finally, we summarise the current status and outline future work. The present paper expands on the rationale and description given in (El-Marakby, 1998).

## 2 BACKGROUND

RTP, the Real-time Transport Protocol, is mainly used for real-time transmission of audio and video over the Internet in multicast and unicast modes (Schulzrinne, 1996). It provides several functions:

- Identification of payload data type to identify the format of the payload data.
- Sequence numbering to detect data packet loss and out-of-order packets.
- Timestamping so that data is played out at the right speeds (Rosenberg, 1997a).

RTCP, RTP's Real Time Control Protocol, is used in monitoring the Quality of Service (QoS) of data delivery and in conveying minimal session control information to all members in an audio/video RTP session.

Both RTP and RTCP are integrated within the application such as the Mbone video and audio applications (e.g. vic and vat).

RTCP has five types of report that are periodically sent to all members in the session. The most important are the feedback reports, namely the Sender Report (SR), and the Receiver Report (RR). SR and RR differ only in that the SR is issued by a receiver which is also a sender whereas the RR is issued by a receiver which is not a sender. Both SR and RR contain performance statistics on the total number of packets lost since the beginning of transmission, the fraction of packet loss in the interval between sending this feedback report and sending the previous one, the highest sequence number received, jitter, and other delay measurements to calculate the round-trip feedback delay time. The SR provides more statistics summarising data transmission from the sender, e.g. timestamps, count of RTP data packets, and number of payload octets transmitted.

The SR and RR have several functions. RRs are used mainly in sender-based adaptive applications. The sender can modify its transmissions dynamically based on the RR feedback it receives from its receivers. The packet loss parameter in the RRs has been used as an indicator of congestion in the network. So, after receiving the RRs from the receivers, the sender may increase or decrease its rate of data transmission according to the packet loss fraction it received within the current interval. This rate adaptation helps to reduce network congestion and adapts to changing network conditions (Bolot, 1994), (Busse, 1996), (El-Marakby, 1997a), (El-Marakby, 1997b). The SR is useful in lip-synchronisation (inter-media synchronisation) and in calculating transmission bit rates. Both SR and RR feedback can be used by a third-party monitor which does not receive RTP data but only RTCP packets. This monitor can be an Internet Service Provider (ISP) or a network administrator. It monitors performance of the network and diagnoses its problems (Schulzrinne, 1996).

The other three types are the Bye report which is used when a member is leaving the session, the Application-defined RTCP packet (APP) report which is used for experimental use with no official packet type registration, and the Source Description (SDS) report which provides identification information about all members in the session.

In the next section, we shall describe some of the RTCP scalability problems. Then we will discuss the functionality of RRs with respect to current requirements

of real-time adaptive applications in the Internet today.

### 3 RTCP SCALABILITY PROBLEMS AND FUNCTION OF RRs

The feedback provided by Internet applications has proved to be useful as no special support is needed from the network to detect its current state. The RTCP feedback is used in adaptive applications as well as in monitoring.

RTCP scales well for small multicast groups but a scalability problem arises when it comes to a group of *thousands* of users. Some of these problems are addressed in (Rosenberg, 1997a), (Rosenberg, 1997b).

#### 3.1 RTCP feedback problems in a large dynamic group

We will explain first how the interval between RTCP packet transmissions is calculated. All RTCP reports multicast to all members in the group must not consume more than a small fraction (nominally 5%) of the whole bandwidth assigned for the session (Schulzrinne, 1996). Hence, every member has to store an estimate of the size of the group by counting distinct RTCP reports sent to the multicast group. Consequently, members scale back their RTCP reporting interval based on the group size they calculated. That is to say, as the group size increases, each member increases its reporting interval and as the group size decreases, every member decreases its reporting interval. As a result, the bandwidth limit for RTCP reports does not exceed 5% of the whole session bandwidth regardless of changes in the group size at any time during the session.

The following are some of the problems encountered when using RTCP feedback in a large dynamic group:

##### *Feedback delay*

The feedback should be sent periodically within acceptable time intervals. In a large RTCP group, this does not happen. Feedback is sent very rarely or not at all. This happens because the RTCP reporting interval grows linearly with the group size. So, as the group size increases, the RTCP interval increases resulting in infrequent RTCP feedback reports which decreases the significance and value of the feedback (Rosenberg, 1997a).

##### *Increasing storage state*

In order to calculate the size of the group, every member has to store a count of distinct members it heard from during the session (Rosenberg, 1997a). So as not to count duplicate members, the unique Synchronisation Source identifier (SSRC) found in the RTP header is stored for every distinct member. Of course, storing all the distinct SSRC identifiers for a large group causes a storage scalability problem for every member.

This problem was discussed in (Schulzrinne, 1997) where a SSRC sampling algorithm is described.

### *Multicasting RRs to the whole group*

RRs are multicast to every member in the session. As mentioned before, RRs are mainly used by the senders for adaptation. So, it seems there is very little benefit having each member send its RRs to every other member in the group which are not senders. In addition, members connected to low bandwidth links would not want part of their bandwidth to be used by incoming RRs when this bandwidth usage would be of little or no advantage to them. Moreover, the processing load at every member may increase because of processing incoming RRs from other receivers. Furthermore, if congestion occurs in the network, it is more likely to affect local members near the congested link. Hence, their RR feedback reports will be more or less similar and hence by decreasing the number of redundant RRs that are multicast, congestion can be reduced (Aboba, 1996).

### *Initial feedback flood*

When a very large number of members join simultaneously (e.g. at the start of a MBone multicast session announcement) (Rosenberg, 1997a), it will not be possible to get an accurate estimate of the group size. Each member's first estimation of the group size is 1, and so all the RTCP reports are sent within a fixed initial interval. Consequently, congestion can occur in the network and especially at low bandwidth links of some members. In addition, the feedback reports sent by other members may be dropped due to congestion. This results in inaccurate estimation of the group size which depends on counting the reports coming from distinct members. Hence, it will take a long time to converge to a fairly accurate estimation of the group size and thus to an appropriate RTCP interval computation.

This problem of *initial* RTCP feedback reports was solved by Rosenberg and Schulzrinne (Rosenberg, 1997a), by applying a reconsideration algorithm. Members listen to other members in the group before sending their *initial* feedback reports. Consequently, the reporting interval is readjusted before sending the first feedback report.

### *Bye flood*

When a RTP member leaves the group, it multicasts a RTCP bye packet to the whole group. The problem occurs if many users leave the group at the same time. As a result, a flood of Bye packets that may congest the network occurs. The problem was fixed in (Rosenberg, 1997b) by applying a Bye reconsideration algorithm.

## **3.2 Functionality of RRs**

The Internet is a heterogeneous network. Network resources are varied throughout, and users can have different capabilities specifically link bandwidth. One of the most important functions of RR feedback reports is their usage in adaptive applications. By using the *packet loss fraction* in RRs, the sender can detect network congestion. Hence, the sender changes its rate of data transmission to adapt to changing network conditions and to help reduce congestion. This technique has proved to be useful for unicast applications. However, for multicast applications in the heterogeneous

Internet, the sender ends up decreasing its data transmission rate to suit the receiver with the lowest capability. Consequently, the sender will not be able to meet the various bandwidth requirements of different receivers in the same multicast group.

To accommodate to this heterogeneous environment and to scale to very large number of receivers, a receiver-based rate adaptation scheme is used (McCanne, 1996). The sender sends the data on separate multicast groups. The groups are ordered such that each provides refinement information over the previous group to give increased quality. The receivers can subscribe to one up to all of these multicast groups according to each one's capabilities and according to the current state of the network. Hence, receivers adapt to congestion by joining or leaving multicast groups.

Nowadays, in the Internet, we see a great movement towards receiver-based adaptation schemes. So, in these applications, the packet loss parameter, which is the most significant parameter in RRs, becomes of less or no significance to the sender as adaptations are performed from the receivers immediately without waiting for the sender to react.

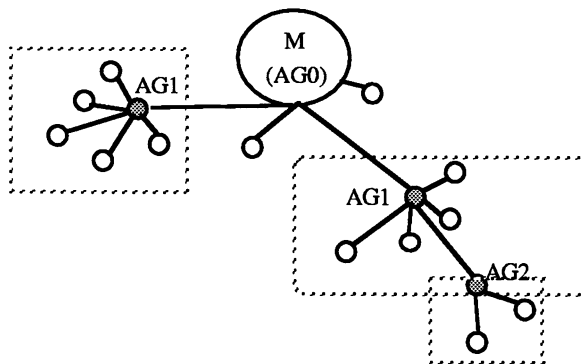
In the next section, we describe the architecture of our scheme.

## 4 OUR SCALABLE RTCP SCHEME

In this section, we describe how locally scoped regions are formed in a hierarchical way. Then, we explain in details the functionality of the Manager.

### 4.1 Overall view

RTCP feedback reports are multicast mainly for receivers to calculate the group size and thus compute their RTCP reporting interval. In our scheme, the members do not need to compute the whole size of the multicast group and RRs are not multicast.



**Figure 1** Structure of our scheme showing members in local regions with an AG (shadowed circle) per region and a Manager (M) at the root of the hierarchy.

Figure 1 depicts the structure of our scheme which organises members



dynamically in a multi-level hierarchy of local regions. Each region has an aggregator (AG). Local members send their RR feedback reports with limited scope to reach their own AG which gathers and aggregates statistics from these reports which it passes to a Manager. The Manager computes additional statistics to evaluate the transmission quality and to estimate the regions which suffer from congestion.

Our scheme makes use of the Time-to-Live (TTL)\* field in the IP header to allow us to build the multi-level hierarchy with locally scoped regions. We are aware of the problems when using TTL scoping with the Distance Vector Multicast Routing Protocol (DVMRP) (Meyer, 1997). We chose to use TTL scoping because it is simple.

## 4.2 Scheme entities:

The following are the entities of our Scalable RTCP (S-RTCP) scheme:

- Member: A member is a sender or a receiver in the same RTP session.
- LAN Aggregator (LAG): The aggregator for a LAN is also a member which represents *only* local members in the LAN. It aggregates RRs from members in its LAN; it then reports to the Manager.
- Aggregator (AG): The AG is also a member, but it also aggregates RRs from members in its local region (i.e. its children); it then reports to the Manager. The children of an AG can be normal members, AGs, or LAGs. Every AG has a level in the hierarchy. For example, in Figure 1, AG1 is an AG of level 1, while AG2 is an AG of level 2 which is a child of AG1.
- Aggregator Report (AGR): This is a new RTCP report of type AGR. Every AG/LAG sends AGRs to the Manager to summarise the quality received by local members during different intervals.
- Manager: This performs some monitoring and diagnosis functions. It receives AGRs. It is also an AG of level 0 (AG0) as it is at the root of the hierarchy. It receives RR feedback from its direct children who are neither AGs nor LAGs, while it receives AGRs from all the other aggregators in the hierarchy. The Manager should be connected to the network through a fast bandwidth link.

The following subsection provides a detailed explanation of the mechanisms of our scheme.

## 4.3 Scheme description

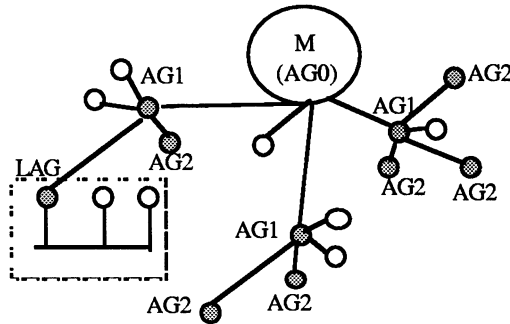
When starting the RTP session, two multicast addresses are announced; the first address is for the delivery of RTP data packets, while the second one is for transporting RTCP control packets. Then, the Manager joins the control multicast group. It receives only the RTCP control packets and not the data packets. It is also the first AG in the multi-level hierarchy (AG0). Afterwards, senders and receivers

---

\* This is an integer field in the IP packet header for constraining the travelling distance of the packet. The source initialises the TTL field with an appropriate initial value according to the distance it wants the packet to travel. Each router decrements this TTL by 1 when the packet arrives. The router discards the packet if the TTL reaches zero.

join the two multicast groups for data and control.

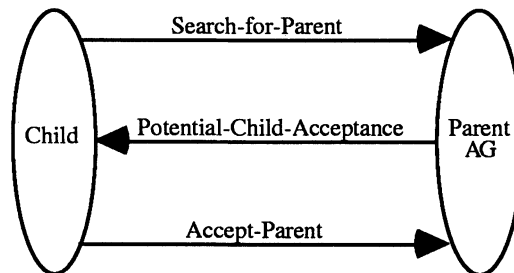
Figure 2 depicts the multi-level hierarchical structure of local regions.



**Figure 2** Organisation of members in local regions where each local region has an AG of a certain level in the multi-level hierarchy. The shaded circles represent AGs, while blank ones represent normal children. Also, a LAG (connected to a LAN) is shown.

### *Selection of Parent AG and formation of local regions*

A new member will perform an expanded ring search (Yavatkar, 1995). The new member will repeatedly search for a Parent AG by increasing the TTL value until it finds a near Parent. First, it multicasts a "Search\_for\_Parent" request (see figure 3) with a small TTL value greater than 1 as it is a well known convention that a multicast packet with TTL=1 is sent only to members in the same LAN. If no reply was received after some time, then it will do another expanded ring search but with a greater TTL value, and so on, until it receives reply(ies) from existing AG(s) of which one will be its future Parent AG.



**Figure 3** Messages interchanged between child and Parent during the process of searching for a Parent.

Each Parent AG stores the current number of its direct children which includes children acting as AGs or normal children (i.e. not AGs). In addition, each Parent stores the maximum number of children, it is allowed to have, initially obtained from the Manager. That is to say, when the very first new members join at the beginning of the session, the only Parent AG by that time is the Manager which is

AG0. If these new members become AGs, the Manager will pass to them these numbers and other information. Then, these AGs might become parents and pass the information they obtained initially from the Manager to their children AGs and so on.

Upon the receipt of the request "Search\_for\_Parent", every *Candidate Parent AG* runs the following tests:

Test 1:

If this *Candidate Parent* can afford to get more children (i.e. it currently has fewer children than the maximum number stored)

then

This new member is considered as a *Potential Child*;

Go to Test 2.

else

It will not send any reply back to the new member;

Exit (i.e. Test 2 is skipped).

Test 2:

If (the level of this *Candidate Parent*  $< N$ )<sup>+</sup> AND

(the distance from this new member  $>$  certain threshold)

then

This member is considered as a *Potential Child AG*.

else

Exit.

The following is a detailed explanation:

- In Test 2, the *Candidate Parent* checks whether its level in the hierarchy is less than  $N$ . If its level is  $N$ , this means that the new member cannot be an AG. Consequently, the height of the tree will be limited. We estimate that a suitable height is  $N=3$ . In addition, the *Candidate Parent* checks if the distance between the new member and itself is greater than a certain value; if greater, then this new member can be an AG, otherwise it will remain to be just a child. This value should not be small as we do not want the new AG to be very close to its Parent AG. This value is passed initially by the Manager to AGs of level 1 which pass it to AGs of level 2 and so on.
- Note also, if an AG belongs to that Parent but does not have any children, then the Parent can replace a new AG instead of that old one which returns to being a normal child. This is not mentioned in Test 2 for simplicity.
- If a LAG wants to join, it will be accepted right away by the Parent AG no matter how many children it has.

After performing these tests at the Parent, if this new member is a *Potential Child*, then the Parent will send a reply "Potential\_Child\_Acceptance". The reply will contain the Parent's IP address and a TTL value of 255. Furthermore, if this new member is a *Potential Child AG*, then the reply will include also the Parent's level in the hierarchy, the maximum number of children this new *Potential AG* will be able to have, the minimum distance allowed between this new member and its future direct AGs, and the minimum and the maximum thresholds for measuring

---

<sup>+</sup>  $N$  is the height of the hierarchical tree. It is equal to the level of the tree + 1.

packet loss (to be explained later).

If there is more than one *Candidate Parent* for this new member, i.e. it receives more than one reply, then it will choose its Parent as the one whose reply carries the largest TTL (i.e. shortest distance from Parent to this new member). That is, as mentioned before, every *Candidate Parent* sends an initial TTL value of 255 in its "Potential\_Child\_Acceptance" reply so that the child can choose that Parent whose reply carries the remaining largest TTL. Note that in the current phase of our work we are measuring the distance in terms of number of hops only and not delay time. The new member will store all the values in the "Potential\_Child\_Acceptance" reply sent by the selected Parent. If more than one reply is received from different Parents, the new member will send "Reject\_Parent" to those Parents not selected.

Then, the new member will unicast "Accept\_Parent" reply to the selected Parent. The Parent will store the remaining value from the original TTL sent by the child in "Accept\_Parent" as an indicator of its distance from that child. Hence, it can detect the distance to the furthest child. In addition, it will increase the number of its children by one.

This restriction of the maximum number of children is an attempt to balance the load of the members among local regions. The Manager is the only AG that does not have this restriction. So if all AGs have their maximum number of children, then any new member will be a child of the Manager. In addition, LAGs do not have restrictions on the number of members in their LANs.

Hence, most members may end up in the vicinity of their nearest AG but this is not always the case.

### *AG leaving or crashed*

Every Parent AG multicasts periodically a local refreshment message to its children with TTL=the stored TTL of the furthest child from it. This message shows that the Parent is still alive and not crashed. If the Parent wants to leave the group, it will multicast locally a Bye packet. Whether the Parent crashed or is leaving, every child will start again the process of searching for a Parent AG through expanded ring searching.

An exceptional case arises when the child is an AG. As mentioned before, every AG can have a maximum number of children. In addition, every AG can accept a maximum number of additional *children AGs* that are not its own children only if their Parent crashes or leaves. Hence, if a Parent crashes or leaves the group, then every child AG of this Parent will search for the nearest Parent. If the nearest Parent can accept more AGs of other Parents, then this child AG will take it as its new Parent, otherwise the child AG will expand its ring searching scope to search for another Parent. Note that this exceptional test was not mentioned before for simplicity.

### *Choice of a LAN Aggregator (LAG)*

If one or more members in the same LAN are participating in the same RTP session, a LAN Aggregator (LAG) is chosen to aggregate information from all the members in the LAN. The process of choosing a LAG depends on scoped multicast

discovery queries to locate a LAG for the LAN.

When a new member in the LAN joins the session, it will send a query packet "Search\_for\_LAG" to search for an existing aggregator for this LAN. If this exists, the LAG will send a reply "LAG\_Exists" that contains its IP address to be stored afterwards by the new member.

If no reply is received within some time, then the new member will consider itself to be the first member in this LAN for this RTP session and will elect itself as the LAG. Then, it starts searching for a Parent AG (see the previous subsections). Afterwards, the Parent AG will pass to the new LAG the minimum and the maximum thresholds for measuring packet loss. These parameters are used when summarising RRs received from members in its LAN.

### *LAG crashing or leaving*

The LAG multicasts periodically "LAG\_Exists" refreshment message to other members in the LAN to inform them that it is alive (Papadopoulos, 1998). If this message is not received within some time, the LAN members will assume that their LAG crashed.

If the LAG leaves the group, it will multicast locally a RTCP Bye message. It will also unicast to its Parent AG that it is leaving.

Whether the LAG crashed or left the group, each member will start the process of choosing a new LAG for their LAN. Each member will try to multicast locally a "Want\_to\_be\_LAG" request. Each member will use a randomised back-off timer and when the timer expires for one of the members, it will immediately multicast locally a "LAG\_Exists" message containing its IP address. Upon receipt of this message, the other members in the LAN will suppress their "Want\_to\_be\_LAG" request and accept this member as their new LAG, then store its IP address. This randomised back-off scheme prevents the flood of the "Want\_to\_be\_LAG" requests if all members multicast at the same time and resolves the problem of choosing a new LAG by directly selecting the LAG whose timer expires first.

### *SR and Bye RTCP reports*

Our scheme deals mainly with the Receiver Reports (RRs). By limiting their travelling distance and summarising important statistics they include, we improve the RTCP scalability. The Sender Reports (SRs) will still be multicast periodically from the sender to the receivers in the session. Note that SRs will not include receiver reporting within them.

Bye reports are sent as follows:

- If a child is leaving, it will send a Bye packet to the Parent AG.
- If an AG/LAG is leaving, it will multicast locally to its children a Bye packet.

## **4.4 Contents of an AGR**

Each AG receives the RR feedback from its direct children which are not AGs or which are AGs with no children. RRs are sent by local members within a certain time interval that is randomised but not to exceed some fixed amount of time. The

AG organises the information, derives some statistics, and includes them into an AGR which reports the quality received by the receivers within a certain interval. Note that the statistics are computed from the RRs of every receiver to a specific sender in the multicast data group.

The QoS statistics and other information contained in an AGR are described below. Some of the functionality of these statistics is explained in the following subsection. The statistics are:

- Number of children that this Parent AG is summarising in the current interval. This number includes only children that send RR feedback.
- Number of children, within the current interval and from the beginning of the data transmission, whose:
  - packet loss exceeds the maximum threshold;
  - packet loss lies between the minimum and the maximum thresholds.
- IP address of the Parent of this AG.
- IP address of the child receiving the worst quality (i.e. which has the highest packet loss in this interval) and the value of packet loss incurred.
- Average, median, and standard deviation of packet loss, in the current interval and since starting transmission, that:
  - surpasses the maximum threshold;
  - lies between the minimum and the maximum thresholds. Note that in order to calculate the median, the AG will sort the packet loss according to the maximum packet loss incurred by every child.

Once all these measurements are computed, they are included in an AGR. Note that these measurements are to evaluate the quality received from one sender. If another sender exists during this interval, then the same measurements are calculated from RRs of local receivers of this other sender. Then, the measurements are appended to the AGR but related to the other sender.

Then, the AG unicasts the AGR to the Manager. In case the aggregator is a LAG, it will send its AGR directly to the Manager too. In the current phase of our work, the AGR is sent directly to the Manager and not to the Parent AG that can pass it to its Parent AG and so on until it reaches the Manager. This is because we do not want to have a long feedback delay until the AGR reaches the Manager.

The following subsection describes the functionality of the Manager.

## 4.5 Functionality of the Manager

The Manager monitors the data distribution in the multicast group and performs some diagnosis functions. It collects and parses the information received from the AGRs during every interval. Then, it logs useful statistics. By making use of information in AGRs, it can estimate whether problems are specific to a certain region or several regions or to all regions of the whole multicast group.

By obtaining the number of children whose packet loss exceeds the maximum threshold as well as the total number of children in the region, the percentage of children suffering from maximum packet loss is derived. As a result, the Manager can pinpoint the regions which are suffering severely from high packet loss. This percentage can turn out to be the same as in another region. However, the case of

one member suffering from maximum packet loss out of a total of 5 members in the region is different than the case of 100 members suffering from maximum packet loss out of 500 in the other region.

Moreover, the mean, the median, and the standard deviation of packet loss that lies between minimum and maximum threshold and that is incurred by all local members, can be used by the Manager. The Manager computes the distribution of packet loss and hence can detect whether packet loss from most members in this range lies nearer to the minimum threshold or nearer to the maximum threshold or in between. The same derivations apply to packet loss greater than maximum threshold.

Every AGR contains the IP address of the Parent of the AG which is sending this AGR. Consequently, the Manager can trace back congestion and detect if it is also spreading in other neighbouring regions (i.e. region of the Parent) or if it is only limited to the current region.

The IP address of the receiver that is suffering from the maximum packet loss might be used by the Manager to launch an *mtrace* between the sender and the receiver to diagnose network problems along the multicast distribution tree and to detect hops showing a significant amount of losses (Thaler, 1997).

In some cases, if some applications still insist on using sender-based adaptive schemes, then S-RTCP can be adapted so that the Manager sends the packet loss value of the receiver suffering most from the highest packet loss incurred in the current interval to the sender. The sender may decrease its rate of data transmission if necessary.

By storing statistics about several consecutive intervals, it can be detected whether the network performance is improving or not.

The estimations mentioned above are derived from short-term statistics (i.e. statistics within an interval). Moreover, similar analysis can be performed on long term statistics to evaluate the quality of the distribution of data during the whole session.

In addition, the statistical data which is gathered and analysed can be used by an Internet Service Provider (ISP), a network administrator, or a technician to estimate the quality received by each region during intervals and during the whole transmission. Furthermore, the ISP can detect the popularity of individual sessions and derive a rough estimate of regions which were densely populated during the whole period of transmission.

The next section presents the benefits of using our scheme.

## 5 BENEFITS OF OUR SCHEME

The following are the advantages we claim of using our scheme in large RTCP groups:

- *Resolving the storage scalability problem:* Members do not store state about every distinct member in the group because they do not need to know the group size.

- *Timely reporting of feedback reports:* Feedback reports become more useful because the RTCP reporting interval does not depend on the group size so feedback delay is minimised. Hence, the experience of members during short intervals of the whole transmission is accurately reported.
- *Effective use of the bandwidth:* Using our scheme, the number of incoming RRs to every member in the group is decreased and there is limited travelling of RRs. This is because of the formation of local regions where RRs are not multicast but are sent with limited scope and not global scope.
- *Decrease in the number of redundant reports:* Even though, in every region we can still have redundant RRs sent to the AG of the region, the total number of redundant RRs, which used to be multicast, is decreased. In addition, measurements in RRs are aggregated into AGRs summarising the quality of the received data.
- *Useful statistics to be used in network diagnosis and in charging:* The aggregated statistics received by the Manager can help a network administrator to diagnose problems in the network. In addition, these QoS measurements can help an Internet Service Provider (ISP) to estimate the quality received in certain regions and the total number of members in the group can show the popularity of individual sessions.

## 6 SUMMARY AND FUTURE WORK

We have presented the problems encountered in the deployment of RTCP in large dynamic groups. Also, we discussed the functionality of RTCP Receiver Reports (RRs) in the current Internet where lots of adaptive applications are using receiver-based rate-adaptive schemes instead of schemes based on sender adaptations. We have designed a Scalable RTCP (S-RTCP) scheme in which members are organised dynamically in local regions; every region has an Aggregator (AG) that receives RRs locally from its members, extracts useful information, derives some statistics, then sends this information to a Manager. The Manager monitors the quality of the data distribution and performs some statistical analysis to estimate which regions are suffering from congestion. We believe our scheme reduces some of the RTCP scalability problems encountered in large groups, namely feedback delay, increase in storage state, and ineffective use of the RTCP bandwidth especially for receivers connected through low bandwidth links. In addition, our scheme directs important information included in RRs to an entity that can make valuable use of them.

In the next phase of our work, we are simulating S-RTCP using the network simulator (NS) (McCanne, 1998) and we will report the results in due course. Also, we intend to investigate more functions that the Manager can do, analyse the limitations of our design, and try to refine it.

## 7 ACKNOWLEDGEMENT

We wish to acknowledge the support of BT Labs for sponsorship of Randa El-Marakby's Ph.D. programme.



## 8 REFERENCES

- Aboba, B. (1996) "Alternatives of Enhancing RTP Scalability", Internet Draft, draft-aboba-rtpscale-02.txt, Nov. 1996.
- Bolot, J. and Turetti, T. (1994) "A rate control mechanism for a packet video in the Internet", Proceedings of IEEE Infocom'94, Toronto, pp. 1216-1223, June 1994.
- Busse, I., Deffner, B. and Schulzrinne, H. (1996) "Dynamic QoS Control of Multimedia Applications based on RTP", Computer Communications, vol. 19, no. 1, pp. 49-58, January 1996.
- El-Marakby, R. and Hutchison, D. (1997a) "Towards Managed Real-time Communications in the Internet Environment", Proceedings of the Fourth IEEE Workshop on the Architecture and Implementation of High Performance Communication Systems (HPCS'97), Greece, June 1997.
- El-Marakby, R. and Hutchison, D. (1997b) "Delivery of Real-time Continuous Media over the Internet", Proc. of the second IEEE Symposium on Computers and Communications (ISCC'97), Alexandria, Egypt, pp. 22-26, July 1997.
- El-Marakby, R. and Hutchison, D. (1998) "Scalability Improvement of the Real-time Control Protocol (RTCP) Leading to Management Facilities in the Internet", to appear in the Proceedings of the third IEEE Symposium on Computers and Communications (ISCC'98), Athens, Greece, June 1998.
- McCanne, S. and Floyd, S. (McCanne, 1998) "LBNL Network Simulator", <http://mash.cs.berkeley.edu/ns>, 1998 version.
- McCanne, S. and Jacobson, V. (1996) "Receiver-driven Layered Multicast", 1996 ACM Sigcomm Conference, pp. 117-130, August 1996.
- Meyer, D. (1997) "Administratively Scoped IP Multicast", Internet Draft, draft-ietf-mboned-admin-ip-space-03.txt, June 1997.
- Papadopoulos, C., Parulkar, G., Varghese, G. (1998) "An Error Control Scheme for Large-Scale Multicast Applications", Proceedings of Infocom'98, San Francisco, CA, pp. 1188-1196, 1998.
- Rosenberg, J. and Schulzrinne, H. (1997a) "Timer Reconsideration for Enhanced RTP Scalability", draft-ietf-avt-reconsider-00.ps, July 1997.
- Rosenberg, J. and Schulzrinne, H. (1997b) "New Results in RTP Scalability", draft-ietf-avt-byerecon-00.ps, November 1997.
- Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V. (1996) "RTP: A Transport Protocol for Real-time Applications", RFC 1889, January 1996.
- Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V. (1997) "RTP: A Transport Protocol for Real-time Applications", draft-ietf-avt-rtp-new-00.ps, December 1997.
- Thaler, D., Aboba, B. (1997) "Multicast Debugging Handbook", Internet Draft, draft-ietf-mboned-mdh-00.txt, March 1997.
- Yavatkar, R., Griffioen, J. and Sudan, M. (1995) "A Reliable Dissemination Protocol for Interactive Collaborative Applications", Proceedings of ACM Multimedia '95, pp. 333-344, 1995.

## 9 BIOGRAPHY

Randa El-Marakby is a Ph.D. candidate at Lancaster University. Randa received a BSc from the American University in Cairo, Egypt and a MSc from University of North Texas, United States, both in Computer Science, in 1988 and 1993 respectively. She has worked for ICL and KPMG Peat Marwick. She is interested mainly in real-time services and protocols for the Internet.

David Hutchison is Professor of Computing at Lancaster University and has worked in the areas of computer communications and distributed systems for the past 15 years. He has completed many UK and European funded research contracts and published over 100 papers as well as writing and editing books on these areas. The main theme of his current research is architecture, services and protocols for distributed multimedia systems. He is involved in several UK and European collaborative projects, in which an integrating theme is Quality of Service for multimedia communications. He is Honorary Editor of the Distributed Systems Engineering Journal, and is a programme committee member for many international conferences and workshops. He has just finished a year's sabbatical leave as a visiting academic at HP Labs in Bristol, UK, at EPFL in Lausanne, Switzerland, and at BT Labs in Ipswich, UK.

## **Part Four**

---

# **ATM Infrastructure**

# Enhanced Convolution Approach for Connection Admission Control in ATM Networks

*J.L. Marzo \*, J. Domingo-Pascual \*\*, R. Fabregat \*,  
J. Solé-Pareta \*\**

*\* Institut d'Informàtica i Aplicacions. Universitat de Girona  
Avda Lluís de Santaló s/n, 17071 Girona (Spain).  
Phone +34 972 418475 FAX +34 972 418399  
e-mail {marzo, ramon}@eia.udg.es*

*\*\* Departament d'Arquitectura de Computadors. Universitat  
Politécnica de Catalunya. Gran Capita, s/n. Modul D6 Campus  
Nord 08071 Barcelona (SPAIN)  
Phone: +34-3-4017001 Fax: +34-3-4017055  
e-mail {jordid, pareta}@ac.upc.es*

## **Abstract**

In this paper the utilisation of the Probability of Congestion (PC) as a bandwidth allocation decision parameter is presented. We assume short buffers at the switch nodes to cope with cell level multiplexation contention ("bufferless" environments). Therefore, delay and cell delay variations are strongly bounded. Moreover, the Cell Loss Ratio (CLR) becomes the critical performance parameter. The validity of PC utilisation is compared with quality of service parameters in bufferless environments. The convolution algorithm is an accurate approach for Connection Admission Control (CAC) in ATM networks with small buffers. However, the convolution approach has a considerable computation cost, in terms of calculation and memory. To overcome these drawbacks, a new method of evaluation is proposed and analysed: the Enhanced Convolution Approach (ECA). In complex scenarios, with ECA, PC calculation can be carried out in real time while maintaining the desired accuracy. Several experiments have been carried out

to compare the demanded bandwidth evaluated by: analytical methods, simulations and measurements in actual ATM switches. The main contribution of this paper is the proposal and analysis of ECA to the PC-evaluation for use in CAC schemes

### **Keywords**

**Connection Admission Control, Traffic Management, B-ISDN and ATM**

## **1 INTRODUCTION**

The Asynchronous Transfer Mode (ATM) transport network is based on fast packet switching using small fixed-size packets called cells. ATM permits flexible bandwidth allocation, so an important objective is to obtain the maximum statistical gain on a shared resource: the physical link. However, the bursty nature of the ATM traffic imposes strict requirements for traffic control. This paper is focused on the utilisation of Connection Admission Control as preventive control method to real time services.

Call Admission Control (CAC) is a procedure responsible for determining whether a connection request is admitted or denied. The procedure is based on resource allocation schemes applied to each link and switching unit. CAC schemes may be classified as non-statistical allocation (peak allocation) and statistical allocation, this paper relates to the second case. In statistical allocation, bandwidth for a new connection is not allocated on the basis of peak rate; rather the allocated bandwidth is less than the peak rate of the source (the sum of peak rates may be greater than the capacity of the output link). The determination of a simple and efficient CAC policy is one of the major challenges in the design and implementation of an ATM-based B-ISDN.

The maximum statistical multiplexing gain can be achieved if the network knows the probability distribution density function of the individual sources. The network needs a complete characterisation of sources with a known behaviour in statistical terms. A set of standardised parameters describes the behaviour of VBR connections in statistical terms. This parameters are: Peak Cell Rate (PCR), Sustainable Cell Rate (SCR) and Burst Tolerance (BT). The VBR bearer capability can be partitioned in two types: a) real-time (rt-VBR) that requires tightly constrained delay and delay variation, (as voice and video interactive applications), and b) non-real-time (nrt-VBR) where only a maximum cell transfer delay is considered (e.g. data transmissions with QOS guaranteed). This paper is mainly focused on CAC aspects relating to (rt-VBR) traffic management.

Adequate traffic characterisation is required to properly design and operate the ATM network, but the wide range of possible future services make this task very complex (Kleinewillinghöfer, 1991). Inevitably, any characterisation of traffic must be in terms of the specific times at which cells are generated by the traffic source.

## **2 BANDWIDTH ALLOCATION IN ATM NETWORKS**

### **2.1 Previous Work**

The exact evaluation of the possible connections onto a link, maximising the statistical multiplexing gain with guaranteed QOS, is a difficult aspect in ATM

networks management (Castelli, 1991), (Hui, 1988), (Bolla, 1997) and (Ohta, 1992). This is due to QOS parameters dependencies: assigned bandwidth and buffer size in a link.

J. Hui proposed a multilevel congestion and control model mechanism in (Hui, 1988). This model defines three different levels: the cell (packet), burst and call level. Those levels are based on the behaviour of the integrated traffic. Different statistical parameters are required to define the traffic at each level.

With reference to the QOS parameter CLR, it can be analysed in both, cell (CLRC) and burst (CLRB) levels. CLRB (burst) is the dominant factor for large buffers and CLRC (cell) is the dominant factor for small buffers (Castelli, 1991), (Handel, 1994) and (Miyao, 1993). It is very interesting to analyse environments with buffers large enough to make CLRC negligible, but small enough to trail the approximation for CLRB close to a bufferless model. This relevant aspect is further detailed in this work.

It is possible to assign an equivalent bandwidth (effective bandwidth for some authors) to each source that reflects its characteristics. The notion of "effective" bandwidth for each connection aims to summarise in a single parameter the bandwidth and QOS requirements of a connection. At the burst level, two different approaches for equivalent bandwidth evaluation are studied by (Guerin, 1991) and (Gallasi, 1989), in which different aspects of the behaviour of multiplexed connections are considered and fluid-flow model and stationary bit-rate distributions are presented. The fluid-flow model is also studied by (Castelli, 1991).

The fluid-flow model estimates the equivalent bandwidth when the individual impact of connections is critical. This model does not consider any multiplexing aspect. The fluid-flow model assumes that the information arrives uniformly during a burst and that the server removes the information from the queue in the same manner. Yang and Tsang in (Yang, 1995) describe an approach to estimate the cell loss probability for traffic scenarios with identical traffic sources (homogeneous traffic).

**Stationary approximations:** In this case the effect of statistical multiplexing is the dominant factor, and it considers that cells are lost when the instantaneous rate is greater than the bandwidth provided by the link. Small buffers are not effective at the burst level. Three methods are introduced below.

**Binomial:** the distribution of the aggregate bit rate on a link can be determined from the stationary distribution of the Markov chain formed by the superposition of sources.

**Gaussian:** this scheme (also referred to as both the normal approximation and the two-moment allocation scheme) assumes the independence of the traffic behaviour of the connections and characterises the multiplexed traffic by a normal distribution. The sum of the means and the sum of the variances of each connection give these parameters. A connection is only accepted if the congestion probability derived from the tail of the normal distribution is less than a pre-specified threshold. The Gaussian assumption is not applicable when there are small numbers of very bursty connections with high peak rates, low utilisation, and long burst periods.

**Convolution:** the exact distribution of the aggregate bit rate can be determined by convolution using the exact bandwidth requirements of each traffic type. This method is based on the formula:

$$P(Y + X = b) = \sum_{k=0}^b P(Y = b - k)P(X = k) \quad (1)$$

X and Y refers to the bandwidth requirement of the new connection and of the already established connections respectively; and b denotes the instantaneous required bandwidth. The above expression allows the evaluation of the distribution function of the demanded bandwidth on a link; this method is explained in detail in section 4.1.

The Decision Criterion in order to accept a new connection X when convolution is used in CAC is based on the Probability of Congestion (PC):

$$PC(Y + X) = P([Y + X] > C) = \sum_{b>C} P(Y + X = b) < \varepsilon \quad (2)$$

**Heuristic Methods:** the other group of CAC approaches is based on heuristics and data modelling techniques (Saito, 1992). The neural network and fuzzy logic based approaches are example of this kind of approaches. Heuristic approaches provide a mechanism for clustering data obtained from ATM traffic measurements in a structure that constitutes the traffic model. A common example is a net structure composed of a set of neurones and respective connections for neural nets and a rule structure composed of a set of “if-then” associations of variables in the case of fuzzy systems (Ramalho, 1994).

## 2.2 Drawbacks

Several limitations have been found in the previous studies: **Inter-dependencies:** all studied models describe the behaviour of the sources without considering their interactions inside the network. The feasibility of performance objectives in ATM networks with correlated traffic is also studied in (Hee, 1993). **Heterogeneous environments:** normally, the performance of the evaluation approximations loses accuracy in heterogeneous scenarios. In these environments there are a trade-off between the Integration (Complete Sharing) vs. Segregation (Complete Partitioning) approaches. **Calculation effort and accuracy:** accurate evaluations have been simplified in order to reduce the complexity of calculations and the required memory, consequently, a reduction of the accuracy is obtained. In (Guerin, 1991), the convolution approach is first applied solely as a binomial distribution over homogeneous sources. Later, the Gaussian distribution is proposed as an approximation to the exact value. **Cell Loss Probability evaluation** (individual CLR): normally, different classes of traffic are segregated to different VPs. Therefore, all individual connections have the same QOS. Nevertheless, it may be more efficient to transport different classes of traffic by the same VP; then, connections on the same VP have different CLR, and, thus, imply different QOS for different classes. This individual CLR<sub>i</sub> for each class of traffic is difficult, or impossible, to obtain.

### 2.3 Experiments of Bandwidth Allocation based on analysis

In the next stage of the work, experiments have been carried out in order to compare the behaviour of different bandwidth allocation approaches for a set of scenarios. Fluid-flow, Linear Approximation and Gaussian are contrasted to the Convolution Approach.

Convolution results are compared to Linear, Fluid-Flux and Gaussian approximations. They have been evaluated in the following manner:

- Linear method: the maximum number of sources that the link can transport  $N_j$  is evaluated by an exact method. The effective bandwidth is  $C/N_j$  for the  $j$ -type sources. Similar procedure is applied to the remaining types. In this case, it is necessary to derive its value off-line; this may be achieved either through analysis or through simulation and experimentation.
- Fluid-Flow model: this approximation has good accuracy when either the number of connections is small or the actual total equivalent capacity is reasonably close to the overall mean rate. An approximation presented in (Guerin, 1991) is used.
- Gaussian allocation approach: this method is based on the assumption that the distribution of the required bandwidth of the existing calls can be approximated by a normal distribution with the same mean and variance. That allows the use of standard approximations to estimate the tail of the bit rate distribution. Formula (6) has been used for this approach.
- Convolution Approach. An exact evaluation of the instantaneous rate on the link is obtained. Utilising the bufferless assumption, Convolution is used for Bandwidth Allocation. In section 4, a detailed study is presented.

All methods calculate a demanded bandwidth in order to ensure a pre-set upper bound for cell loss ratio.

Homogeneous traffic experiments: for these experiments a set of 50 On-Off sources has been analysed. These sources have a mean burst period equal to 100 ms; and the maximum PC allowed is  $10^{-5}$ .

Fig. 1 shows the demanded bandwidth (y-axis) evaluated by both: Fluid-Flow and Convolution. The number of sources is 50; and each connection has a peak rate equal to 4 Mb/s. The utilisation of each source varies from 10% to 80% (x-axis). The fluid-flow model is evaluated for different buffer sizes ( $b = 0.01, 1, 2$  and 3 Mbit),  $b = 0.01$  means in fact small buffers, no differences have been obtained for buffer size up to 128 cells.



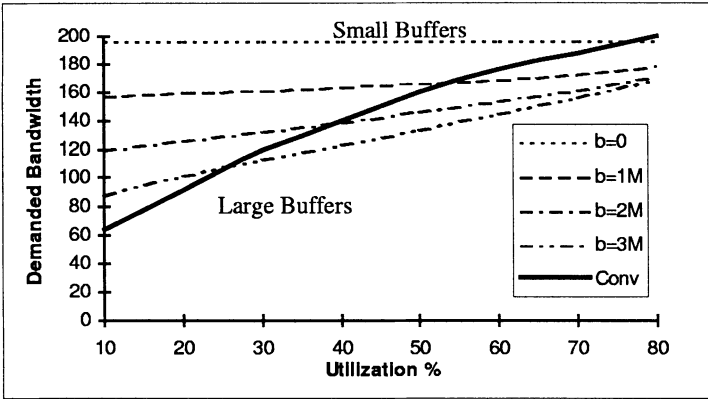


Fig. 1 Demanded Bandwidth vs. utilisation

For source utilisation higher than 25% and with a buffer size of 3 Mbit the required capacity evaluated by convolution is greater than the required for the Fluid-Flow approximation. Furthermore, when the size of the buffer is 1 Mbit, the same effect occurs when the source utilisation is higher than 55%. When using small buffers, the convolution approach always gives more accurate results. Note that small buffers are used in our study to limit maximum cell delay and jitter. Consequently, Fluid-Flow approximations will not be taken into account in the following experiments.

On the other hand, the convolution approach always evaluates a more accurate demanded Bandwidth than the Gaussian model. From 10 % to 40 % over-estimation is observed with the Gaussian model compared to Convolution, for 80 % of source utilisation.

Heterogeneous traffic experiments. Several experiments with mixed traffic are now presented. On the y-axis the demanded bandwidth is shown and on the x-axis all the possible combinations of traffic are presented. The source characterisation has been chosen in the GMDP model. The following table shows the characteristics of the used sources.

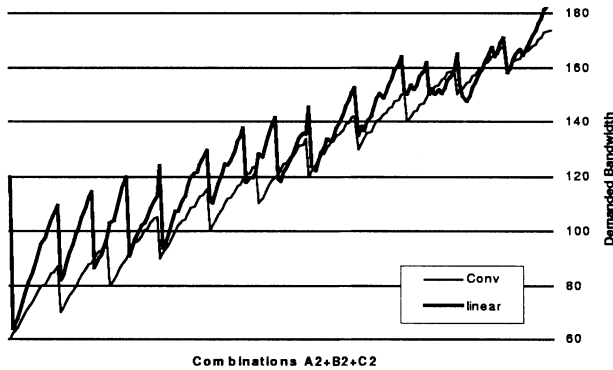
	State 0		State 1	
	Rate (Mbits)	Prob	Rate (Mbits)	Prob
A2	0.4	0.625	2	0.375
B2	2	0.625	10	0.375
C2	6	0.625	30	0.375

Table 1 Source description

When two types, B2 and C2, are mixed, and the maximum number of connections is 7 and 42 respectively. In this scenario the Gaussian approximation is also

accurate, except for combinations of traffic that correspond to a small number of connections.

In the following experiment, three types are mixed: A2, B2 and C2, the maximum number of connections is 15, 10 and 3 respectively.



*Fig. 2 Demanded Bandwidth for heterogeneous traffic*

For simplicity, only combinations with one C2 connection and varying B2 and A2 types of traffic are plotted Fig. 2, since the remaining combinations show similar behaviour. The linear approximation has a changing behaviour that tends to be conservative. Differences up to 50%, case of low number of connections, can be observed.

Using small buffers, a premise in this study, the convolution approach gives always more accurate results than Fluid Flow approximations. The required capacity evaluated by convolution is normally less than the one required for the Gaussian assumption. In the presence of bursty traffic, utilisation less than 20%, the Gaussian evaluation is too optimistic. Guerin et al. (Guerin, 1991) propose using Fluid-Flow approximations in this situation. Moreover, for small buffers the required bandwidth is perceptively higher than the Gaussian. Moreover, it is not clear how to define this situation. This effect is unwanted because the actual cell loss may be greater than the evaluated cell loss ratio and, consequently, the QOS requirements established for the user could not be guaranteed. Finally, the linear approximation varies from pessimistic to optimistic depending on the mixture of traffic.

### 3 THE PC AS BW ALLOCATION DECISION PARAMETER

#### 3.1 ATM network model

There are some services for which the QOS has real time constraints (i.e. interactive services), for delay; these services are considered in the standards as rt-VBR bearer capabilities. Therefore, very large buffers cannot be introduced and buffer dimensioning is carried out taking into account the cell level contention. Also, suitable buffer sizes can be selected to ensure that the maximum cell delay is less than a pre-specified limit (Yang, 1993). Under that premise, Cell Loss Ratio (CLR) is the major relevant Parameter of QOS.

Therefore, the buffer size of the statistical multiplexer is assumed to be small, in order to guarantee an acceptable maximum delay (e.g. 50 cells corresponds to 140  $\mu$ s., link rate = 150 Mbit/s). LAN interconnect is an important example of a service which requires both a low loss rate and a low end-to-end delay (Wright, 1989) due in the main to time-outs in the LAN protocols.

### 3.2 The Probability Of Congestion.

To work with small buffers implies that the traffic bursts cannot always be saved in the buffer. Therefore, the burst length is irrelevant because the majority of cells will be lost. On the other hand, it is not likely that users will be able to supply information about the burst length at connection set-up.

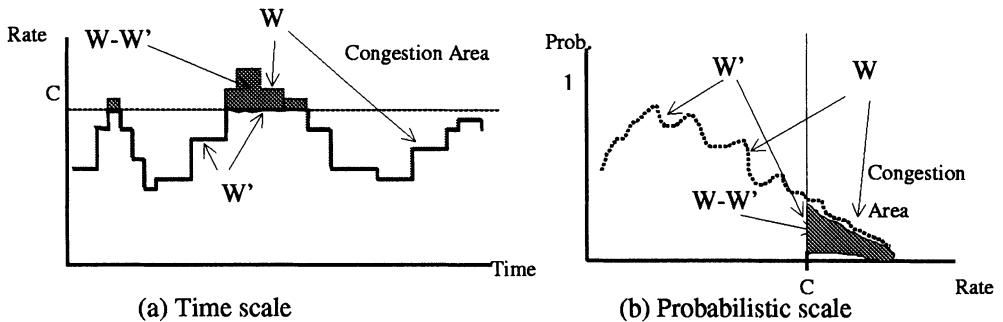


Fig. 3 Congestion in an ATM link

$W$  and  $W'$  represent the offered and the carried traffic respectively. Under the bufferless assumption  $W-W'$  means the lost traffic. Fig. 3 (a) shows the instantaneous aggregated rate of all sources connected against time. Fig. 3 (b) shows the probability associated to a given instantaneous aggregated bit rate of all sources; all situations corresponding to rates greater than  $C$  (at right of  $C$ ) are in a congestion state.

The probability of congestion ( $PC$ ) is the sum of probabilities corresponding to rates greater than  $C$ , which is the shadow area. The probability of congestion does neither state how many cells are lost nor the duration of the congestion state, but only that there is cell loss (Iversen, 1991). The load region admissible is approximated using parameters such as; the mean load, the congestion probability, and the ratio of cells exceeding the link capacity for the total cell stream and for each individual connection. For arbitrary mixes, the load of the link may provide only little information about the cell loss probabilities. Cell losses are quite likely if the bandwidth required at the burst level exceeds the capacity of the link. These events are taken into account by the  $PC$ .

$$PC(Y) = P(Y > C) = \sum_{R > C} P(Y = R) \quad (3)$$

$Y$  means the bandwidth distribution in terms of instantaneous rate, and  $C$  is the rate of the link. However, the  $PC$  does not give any information about the number of cells lost in case of congestion unlike Cell Loss Ratio (CLR). In a short congestion

state, all cells may be buffered with no cell losses occurring. Nevertheless, when a burst's duration is longer than the size of the buffer, then almost all cells exceeding the link capacity are lost. In this case, the relation between PC and CLR is approximated by:

$$CLR(Y) = \frac{\sum_{R>C} (R - C)P(Y = R)}{E(Y)} \quad (4)$$

If the buffer size is sufficient for cell contention, the evaluated CLR provides an upper bound to the total cell loss probabilities. The PC model is a stationary approximation, in other words, a probabilistic scheme.

#### 4 BANDWIDTH ALLOCATION BASED ON THE PROBABILITY OF CONGESTION

In this section, different methods to obtain Probability of Congestion (PC) on a link based on the convolution function are presented. The cost involved in the PC calculation is also analysed.

##### 4.1 The Formula-Based Convolution Approach

This section contains the calculation of the bandwidth requirements of the superposition of several sources. This approach is based on the well-known expression of the convolution procedure denoted by:

$$Q = Y * X \quad (5)$$

which is evaluated by the following expression:

$$P(Y + X = b) = \sum_{k=0}^b P(Y = b - k)P(X = k) \quad (6)$$

where Q is the bandwidth requirement of all established connections including the new connection; Y is the bandwidth requirement of the already established connections; X is the bandwidth requirement of a new connection, and b denotes the instantaneous required bandwidth. This function is expressed as the probability that all traffic sources together are emitting at a given rate b. We take into account that the evaluated offered load is not the link load itself, but the load generated by all the traffic sources intended to be carried by the link. The link carries this load in non-congestion state only.

The direct application of the expression (6) in order to evaluate the convolution is difficult in practice. In order to obtain the probabilistic distribution on the link, a vector containing all possible rate-probability pairs is defined; this vector is called System Status Vector (System-SV). To obtain the complete System-SV, the following process is carried out: whenever a new connection demand arrives the System-SV must be updated; the corresponding Source-SV is used to do this update, and for each old System-SV element a set of new System-SV elements is generated. The rate of each new element is the sum of the existing rate and the rate corresponding to the state of the new source. The probability of each new element

is the product of the existing probability and the new probability corresponding to the state of the new source. By using this method N-1 convolutions are needed for each new connection. The expression (5) can be re-written as (Iversen, 1990):

$$Q_n = Q_{n-1} * X_n ; n \in \{1, 2, \dots, N-1\} \quad (7)$$

Considering  $Q_0 = X_0$ . Clearly, we should carry out N-1 convolutions to obtain the global distribution. At any point in time, an ATM link can carry several thousand connections. As described in a previous section, when a connection is accepted, the new connection is convoluted with the global steady-state probabilities of all existing connections. When a connection terminates it would be preferable to deconvolute all existing connections. So the feasibility of the deconvolution operation is very important. The problem is that the global steady state has now changed; this means that some previously calculated values are lost. The reason for this is that a) the accumulation of probabilities, corresponding to the same rate and b) very small intermediate values, are not considered. Furthermore, by not truncating the state, the space required for storage increases; the number of arithmetic operations further increases. These aspects, relating to accuracy and cost, are more widely developed in the following sections.

### Drawbacks

In (Iversen, 1990 and 1991), (Kroner, 1990), (Kaltenmorgen, 1992), (R1022, 1990), (Del 122, 1991), (Miah, 1994) and (Ramalho, 1994) some limits of the Convolution Approach are pointed out. **High cost in terms of storage requirements.** Note That a huge amount of memory storage M is required by the System-SV. This requirement increases dramatically with the number of connections N and source states  $T_j$ . **High cost in terms of calculation.** The computing time depends on the complexity of the distribution itself. The time needed for the convolution increases with the number of states per connection. **Individual QOS.** The evaluated link status using a convolution approach makes no distinction between individual connection. Thus, the individual QOS, in terms of cell loss, for each type of source is not available.

### 4.2 The Enhanced Convolution Approach (ECA)

To overcome the drawbacks associated with the PC calculation, a new method of evaluation is proposed: the Enhanced Convolution Approach (ECA). In this method, the multi-nomial distribution function is first applied to groups of the same type of sources, and the global state probabilities are finally evaluated by convolution of the partial results obtained from the different existing groups of sources.

The state of the link can be expressed as a function of the number of active connections of each service type ( $n_0, n_1, \dots, n_j, \dots, n_{L-1}$ ). This is because the state of the link depends only upon a service's occupancy. First the multinomial function is applied to homogenous sources producing intermediate results. Finally, from these intermediate results a final result is obtained by convoluting one element of a given class of traffic with one from each of the other classes. This process is called *multi-convolution* in this study.

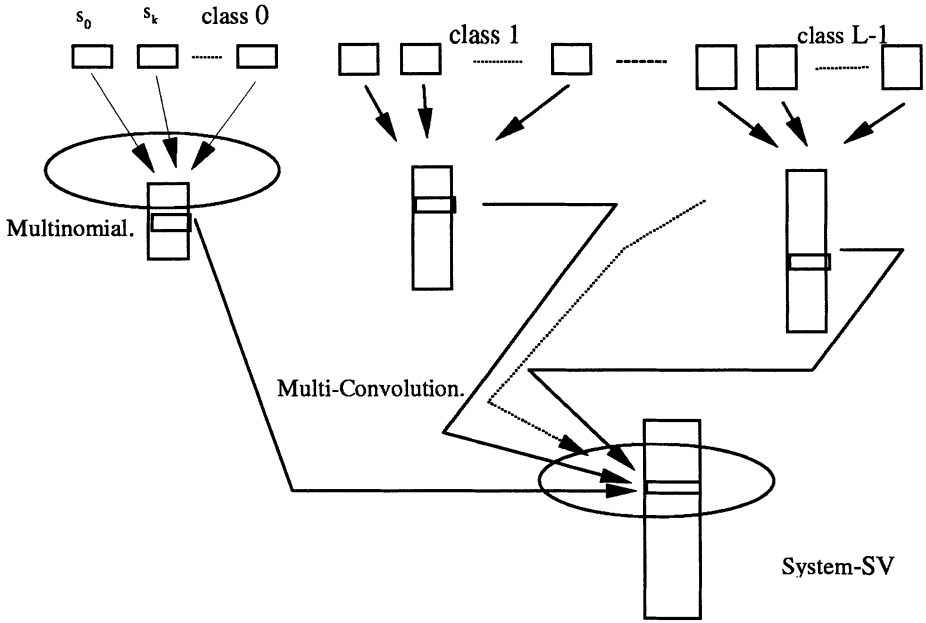


Fig. 4 Overview of the method.

### The Multinomial Distribution Function (MDF)

After computing the convolution the same rate may appear more than once in the System-SV. Which elements are repeated? How many times? This is expressed as a probability corresponding to the MDF, see (Ash, 1969) and (Hogg, 1989).

$$P(n_0, n_1, \dots, n_{T-1}) = \frac{N!}{n_0! n_1! \dots n_{T-1}!} p_0^{n_0} \cdot p_1^{n_1} \dots p_{T-1}^{n_{T-1}} \quad (8)$$

The corresponding rate can easily evaluated by the following expression:

$$r(n_0, n_1, \dots, n_{T-1}) = \sum_{i=0}^{T-1} n_i \cdot r_i \quad (9)$$

Note that the probability of each source being in state  $s_i$  is independent of the probability of the other source states.

To evaluate the ECA, some data structures are necessary at this phase. For N connections of the same type, there is an associated Sub-Matrix (SMX).

$$SMX(N) = \begin{bmatrix} n_{0,0} & n_{0,1} & n_{0,j} & n_{0,T-1} \\ \dots & \dots & \dots & \dots \\ n_{r,0} & n_{r,1} & n_{r,j} & n_{r,T-1} \\ n_{M-1,0} & n_{M-1,1} & n_{M-1,j} & n_{M-1,T-1} \end{bmatrix} = \begin{bmatrix} SMX_0 \\ \dots \\ SMX_r \\ \dots \\ SMX_{M-1} \end{bmatrix} \quad (10)$$

$SMX_r$  is the generic row of  $SMX$ . The number of columns is equal to the number of source rates  $T$ .  $SMX$  stores the distribution of the connections from each state. The system load density function is obtained directly from the sub-matrix using the MDF expression.

### The multi-convolution procedure

When there are different types of sources  $j$  (heterogeneous traffic), it is necessary to 'convolute' between all source types. To store all possible combinations relating to the system state, a System Status Matrix (SSM) is defined. The generic elements of the SSM, namely the general system status rows  $SSM_r$ , are generated each by concatenating all possible combinations between the different sub-matrices rows  $SMX_r$ , associated with the  $L$  different  $j$ -types of sources ( $j = 0, 1, \dots, L-1$ ).

$$SSM_r = \langle SMX_{r_0,0}, \dots, SMX_{r_{L-1},L-1} \rangle \quad \forall r=0, \dots, M_{j-1} \quad (11)$$

and from (12)

$$SSM_r = \langle n_{r_0,0}, \dots, n_{r_0,S_j-1}, \dots, n_{r_{L-1},L-1}, \dots, n_{r_{L-1},L_j-1} \rangle \quad (12)$$

Based on the ECA algorithm, grouping connection in types, the following expression for the cell loss probability of the type- $j$  traffic proposed:

$$CLR_j = \frac{\sum_{w>c} \frac{W_j}{W} (W - C) P(Y = W)}{E(Y_j)} \quad (13)$$

$W_j$  is the rate offered by all type- $j$  traffic when the instantaneous offered rate on the link is  $L$  and  $E(Y_j)$  is the mean rate of all traffic of type- $j$ . Both terms are easily obtained during the evaluation of PC based on the ECA algorithm. This is the evaluation of the CLR to a type- $j$  traffic.

This method is widely detailed in (Fabregat, 1995) and (Marzo, 1993).

## 5 CAC BASED ON THE PROBABILITY OF CONGESTION

This section is focused on reasonable real-time processing, storage requirements as well as the maximisation of statistical multiplexing gain. Obviously, all the methods studied intend to guarantee QOS for all established connections.

### 5.1 Implementation Issues

In order to evaluate the PC, it is not necessary to completely evaluate the entire distribution of the instantaneous rate. Using ECA it is possible to evaluate a part of the statistical distribution. The techniques presented next are attempted to evaluate

only the major relevant part of the system state: Congestion. All five cut-off improvements can be implemented simultaneously.

**Link Capacity Cut-off:** the calculation of probability is only carried out in cases where the associated rate exceeds the bandwidth provided  $C$ . The associated probabilities with rates smaller than  $C$  are not calculated (i.e. the MDF is not evaluated).

**Probability of Congestion Cut-off.** The PC of the system is compared with a previously set value in order to guarantee the specific QoS. Therefore, if during the process of calculation the current PC exceeds a pre-set value, the process is stopped and the calculation cost is thus reduced.

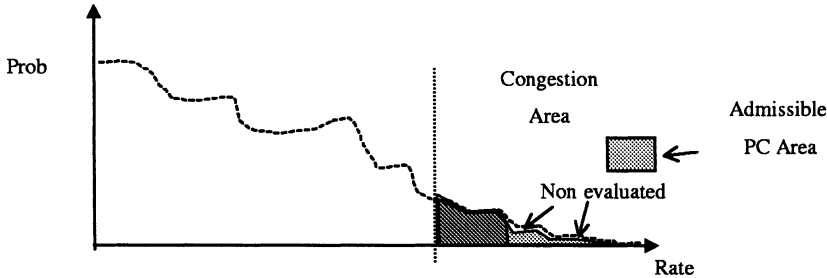


Fig. 5 Probability of Congestion Cut-Off

**Partial Sorting Cut-off.** Furthermore, in each System Status Matrix (SSM) the rows generated are not examined in an arbitrary order, but are graded according to rate, so when the pre-set minimum rate  $C$  is reached, the process is terminated and a further saving computation time is achieved.

**Small probabilities Cut-off.** When the probability obtained by the Multinomial Distribution Function, is less than a pre-set threshold value this result can be ignored. Therefore, the corresponding element is not stored. This threshold value depends on the maximum congestion probability and on the number of connections.

**Grouping states Cut-off.** For those classes with a large number of connections the majority of information enclosed in the sub-matrix may be summarised by grouping states. This mechanism could be applied independently to each class of sources before evaluating the second phase in the calculation of PC.

## 5.2 Cost Experiments

All experiments have been based on a General Modulated Deterministic Process (GMDP) source model. The GMDP model describes the behaviour of a traffic source at cell and burst level. The number of states for  $j$ -type sources is  $S_j$ . In each state  $i$  (with  $i = 0, 1, \dots, S_j-1$ ), during the corresponding sojourn time  $SojT_{j,i}$ , cells are sent with regular inter-arrival times (constant rate  $r_{j,i}$ ).

The evaluation cost is measured in terms of a number of different metrics: **Storage** requirements are measured in elements, each element has to store a rate and a probability. The **time** parameter corresponds to CPU time and is expressed in normalised time (seconds in the presented experiments). **Sorting** techniques are required to put the partial status vector in order. The quick sort algorithm is used



when necessary. The cost is expressed as  $x \cdot \log(x)$ , where  $x$  is the number of elements to be arranged. Calculation cost is expressed as a normalised combination of additions and products

The computational efficiency has been measured for both the formula based (basic) convolution and the ECA. A comparison in time is shown in the 'speed-up' column of the following table, the time obtained for the basic convolution is set to 1; the evaluations for the enhanced convolution approach are normalised to this value.

Two types of sources (60 and 75 connections) are multiplexed in an 600 Mbit/s ATM link. In the presence of only one class of traffic the application of the ECA increases the speed-up factor.

	Mval/Gr	PC	CLP	time	sp-up	storage	sorts	cost
Basic		2E-4	1.2E-5	4.86	1	13252	17480	162
ECA	-	2E-4	1.2E-5	0.95	5	3439	5	9398
ECA	1.E-7/-	2E-4	1.1E-5	0.04	122	709	1	327
+	-/3	"	"	0.12	41	1147	5	1095
Cut off	-/5	2E-4	1.2E-5	0.06	81	689	5	432
	1E-8/3	2E-4	1.2E-5	0.02	243	372	1	97
	1E-08/5	2E-4	1.2E-5	0.01	486	163	1	70

*Table 2 Cost results for heterogeneous traffic*

Sorting is the dominant factor for the formula-based convolution, whereas cost evaluation is the dominant factor for the enhanced convolution. Another conclusion is the efficacy of the small probabilities cut-off. The first direct implication is the reduction in the storage requirements. Moreover, this reduction in the intermediate vectors implies a rise from 5 up to 500 times faster in the carried out experiments.

## 6 CAC EXPERIMENTS

This Section discusses different aspects of the behaviour of cell streams in an output buffer corresponding to an ATM link and is illustrated by experimentation. CAC experiments relating to Fuzzy logic and (M+1)-MMDP approaches are presented. Measurements in an ATM test bed are also included.

The experiments described in this section refer to a single ATM link and the QOS is expressed in terms of cell loss at the output buffer of an ATM switch. The traffic sources are VBR sources, modelled as On-Off sources described by the peak and mean bit rates and mean burst length. Experiments in homogenous scenarios reveals similar results than for heterogeneous scenarios and have been omitted in this paper.

RACE projects provide ATM test-beds on which measurements and tests can be performed (Kuhn, 1994). This set of experiments enables comparison of the average cell loss results obtained from on-line measurements in the Exploit ATM test-bed in Basel (R2061/28, 1994) with the cell loss predictions given by both the ECA and FCAC approaches for homogeneous and heterogeneous traffic scenarios

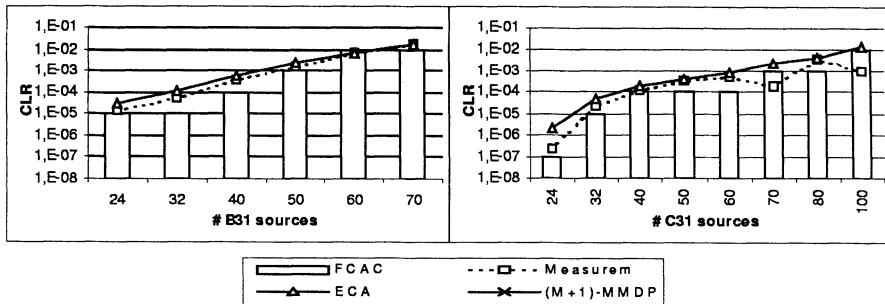
on a single ATM link. Although FAC attempts to predict the maximum cell loss ratio per connection instead of the average cell loss ratio for the aggregate traffic, for the sake of comparison with the results obtained from on-line measurements, FCAC was trained to predict the average cell loss ratio instead. In Table 3 the traffic sources used for the comparison experiments are described. The link capacity considered is 155.52 Mbit/s and the output buffer size is 27 cells.

Traffic type	Peak Rate (Mbit/s)	Mean Rate (Mbit/s)	Mean Burst L. (cells)
A31	31.1	6.22	1467
B31	7.78	3.89	917
C31	1.94	0.97	229

*Table 3 Characteristics of the traffic sources*

4 A31 connections are mixed with 6 to 24 connections of B31 traffic in experiment B5. 4 A31 connections are mixed with 24 to 100 connections of C31 traffic in experiment B6. We can see in both experiments, that the cell loss results obtained by ECA and FCAC are very similar to those obtained by measurements in the Exploit test-bed.

Considering that the buffer size used in the ATM test-bed experiments is small (27 cells), it is not surprising that the cell loss predicted by the ECA is so close to the cell loss measured values. This also explains the slight difference between the measurements curve and the convolution curve as some of the generated cells can be stored in the server output buffer, and, therefore, a more optimistic cell loss is obtained (see measurements curve).



*Fig. 6 (a)Experiments B5 and (b) experiments B6*

The prediction results based on the convolution approach have been obtained using the ECA algorithm without considering optimisations. The cut-off mechanisms and other improvements biased to achieve a fast evaluation of the CLR can only be used for CAC. Anyway, for the experiments presented, the time required for the evaluation was negligible.

## 7 CONCLUSIONS

In this paper the utilisation of the Probability of Congestion (PC) as a bandwidth allocation decision parameter has been presented. The validity of PC utilisation is compared with QOS parameters in small buffer environments when only the CLR parameter is relevant.

To overcome the drawbacks of the formula-based Convolution Approach, a new method of evaluation is analysed: the Enhanced Convolution Approach (ECA). Sorting is the dominant cost factor for the formula-based convolution, whereas calculation is the dominant cost factor for the ECA. With reference to the cut-off mechanisms presented, the major conclusion is the efficacy of the low probability cut-off. This mechanism implies a reduction in the storage requirements and a reduction of the evaluation time. The ECA also enables the computation of the Individual Cell Loss Ratio for each j-class of traffic.

It can be summarised that the convolution algorithm seems to be a good solution for CAC in ATM networks with relatively small buffers. If the source characteristics are known actual cell loss ratio can be accurately estimated. Furthermore, this estimate is always conservative, allowing the provision of the network performance guarantees. We can also conclude that by combining the ECA method with cut-off mechanisms, utilisation of ECA in real-time CAC environments as a single level scheme is possible.

Source modelling for more realistic traffic is now an open issue. The ECA utilisation does not take account of temporal references (burst length), so that the source parameterisation is simplified. On the other hand, more realistic traffic, such as VBR video sources, can be modelled as sources with more than two associated states. ECA can also be use with these new models.

By simple analysis of the ECA evaluation (see Fig. 4), a parallelisation of the ECA algorithm is a further step (the evaluation of each sub-matrix corresponding to each class of traffic can also be obtained in parallel).

## 8 REFERENCES

- Ash, R. (1969). Basic Probability Theory. John Wiley & Sons.
- Bolla, R. Davoli, F. and Marchese, M. (1997) Bandwidth Allocation and Admission Control in ATM Networks With Service Separation. IEEE Communications Magazine. Vol 35 No. 5 130-137.
- Castelli, P. Cavallero, E. And Toniatti, A. (1991) Policing And Call Admission Problems in ATM Networks. Teletraffic and Datatraffic, ITC-13 847-852.
- Del. 122 (1991). Progress Report on the CAC and Policing Experiments. COST.
- Decina, M. And Toniatti, T. (1992) Bandwidth Allocation and Selective Discarding for VBRV and Bursty Data Calls in ATM Networks. International Journal of Digital and Analog Communications Systems.
- Marzo, J.L. Domingo, J. Fabregat, R. and Sole-Pareta, J. (1995) Dynamic Routing Based on a Single Parameter: Link Congestion Probability. High Performance Networking VI. IFIP. Chapman & Hall. 307-318
- Gallasi, G. Rigolio, G. and Fratta, L. (1989). ATM: Bandwidth Assignment and Bandwidth Enforcement Policies. Globecom 89.

- Guerin, R. Ahmadi, H. and Naghshineh, M. (1991) Equivalent Capacity and its Application to Bandwidth Allocation in High-Speed Networks. *IEEE Journal On Selected Areas In Communications*, Vol 9, N7, 968-98.
- Hogg, R. Allen, V. And Craig T. (1989) *Introduction to Mathematical Statistics*. Maxwell MacMillan Int Ed.
- Handel, R. Huber, M. N. And Schroder, S. (1994) *ATM Networks. Concepts, Protocols and Applications*. Addison-Wesley.
- Hee-Jeon, Y. and Viniotis, I. (1993) Feasibility of Performance Objectives in ATM Network Nodes with MMPP Arrival Processes. *Modeling and Performance Evaluation Of ATM Technology (C-15)*. 119-134.
- Hui, J. Y. (1988) Resource Allocation for Broadband Networks. *IEEE Journal on Selected Areas in Communications*, Vol 6 N 9, 1598-1608.
- Iversen, V.B. and Yun Liu. (1990) The Performance of Convolution Algorithms for Evaluating the Total Load in an ISDN System. Ninth Nordic Teletraffic Seminar, Norway.
- Iversen, V.B. and Bohn-Nielsen, A. (1991) Statistical Multiplexing In ATM-Networks. RACE 1022.
- Kaltenmorgen, B. (1992) FIDBP. Connection Acceptance Control. BAF-Project-FIDBP-92301-Cd-Cc/A.
- Kleinvewillinghöfer-Kopp, R. and Kaltenmorgen, B. (1991) Connection Acceptance Control In ATM Networks. Studie des Forschungsinstituts der DBP Telekom, Forschungsbereich 5 'Vermittlung und Netze'
- Kroner, H. (1990) Algorithms For Call Acceptance Control Published At ICC'89-Basic Proprieties And Preliminary Study. RACE 1022. UST\_123\_026\_CD\_CC.
- Kuhn, P.J. (1994). *Performance Modelling and Traffic Engineering For Broadband Communication Networks*. North Holland Broadband Comm.
- Marzo, J.L. Fabregat, R. Domingo, J. and Sole-Pareta, J. (1993). Fast Calculation Of The Cac Convolution Algorithm Using The Multinomial Distribution Function. UK TT Symposium. Performance Engineering In Telecommunications Networks. BT Laboratories Ipswich UK.
- Miah, B. and Scharf, E. (1994) A Real Time Management System for the Two-Level CAC Algorithm. 11 UK Teletraffic Symposium. Cambridge.
- Miyao, Y. (1993) Bandwidth Allocation in ATM Networks that Guarantee Multiple QOS Requirements. ICC93, 1398-1403.
- Ohta, S. and Sato, K. I. (1992) Dynamic Bandwidth Control of the Virtual Path in an Asynchronous Transfer Mode Network. *IEEE Transaction on Communications*, Vol. 40, N°. 7.
- R1022. (1990) Updated Results of Traffic Simulation of the Policing Experiment. Technology For ATD.
- Race 2061 (1994) Exploit, Results Of Experiments On Traffic Control Using Real Applications. Deliverable 28.
- Ramalho, M. F. and Scharf, E.(1994). Fuzzy Logic Based Techniques for CAC in ATM Networks. 11 UK Teletraffic Symposium. Cambridge.
- Saito, H. (1992) CAC in an ATM Network using Upper Bound of Cell Loss Probability. *IEEE Transactions on Communications*. 40, 1512-1521.

- Wright, D. J. and Michael T. (1989) A Characterization of Telecommunication Services in the 1990's. Infocom 89. 624-631.
- Yang, T. and Li, H. (1993) Individual Cell Loss Probabilities and Background Effects in ATM Networks. IEEE ICC33, 1373-1379.
- Yang, T. and Tsang, H. K. (1995). A Novel Approach to Estimating the CLP in an ATM Multiplexer Loaded with Homogeneous On-Off Sources. IEEE Transaction on Communications.

## 7 BIOGRAPHY

Jose-Luis Marzo is an Associate Professor of Computer Science and Communications at the Universitat de Girona. He received the engineering degree in Computer Science (1989) and the Ph.D. degree (1997). Since 1988 he is teaching at the Electronics and Computer Architecture Department. His research topics are Broadband Communications and Applications. Since 1993 he has participated in Spanish Broadband projects. He is IEEE member.

Jordi Domingo-Pascual is an Associate Professor of Computer Science and Communications at the Universitat Politècnica de Catalunya in Barcelona. There, he received the engineering degree in telecommunications (1982) and the Ph.D. Degree in Computer Science (1987). Since 1983 he is teaching at the Computer Architecture Department. His research topics are Broadband Communications and Applications. Since 1988 he has participated in RACE projects, Spanish Broadband projects, and ACTS projects. Since 1995 he is a researcher at the Advanced Broadband Communications Center of the University (CABA).

R. Fabregat is an Associate Professor of Computer Science and Communications at the Universitat de Girona. He received the engineering degree in Computer Science (1989). Since 1990 he is teaching at the Electronics and Computer Architecture Department. His research topics are Broadband Communications and Applications. Since 1993 he has participated in Spanish Broadband projects.

J. Solé-Pareta. Josep Sole-Pareta received his Master's degree in Telecommunication Engineering in 1984, and his Ph.D. in Computer Science in 1991, both from the Universitat Politècnica de Catalunya. Since 1992 he is an Associate Professor with Computer Architecture Department. He has participated in the R&D Spanish Program for the development of the Broadband Communications. Within the ACTS program, he is participating in MICC, IMMP and INFOWIN projects. His research interests are in B-ISDN, ATM Networks and Personal Communication Systems. He is member of the IEEE and the ACM (Sigcomm).

**ACKNOWLEDMENTS:** This work has been supported by CICYT (Spanish Education Ministry) under contract TIC95-0982-C02-02 and TEL97-1054-C03-03.

# **Fast Rerouting in ATM Networks: Pro-Active Search protocol**

*I. Lievens, T. Cattryse, P. Demeester*

*Department of Information Technology, University of Gent*

*Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium*

*Tel: +32 9 264 33 16, Fax: +32 9 264 35 93*

*ilse.lievens@intec.rug.ac.be*

## **Abstract**

This paper introduces a rerouting algorithm for ATM networks - called the Pro-Active Search Protocol - which enables efficient rerouting in a simple and uncomplicated way. Therefor the protocol performs as many tasks as possible in advance to reduce the complexity. Basically this comes to working with shortest path algorithms based on a known network topology. In order to increase the rerouting performance this is combined with real-time actions after a rerouting trigger, by collecting accurate information about the network load. This is achieved by means of a simplified distributed approach.

## **Keywords**

**Network Availability/Reliability, Protocol Design**

## **1 INTRODUCTION**

In ATM Broadband Telecommunication Networks a wide range of services, each with particular demands and characteristics, will be offered to the user. An important issue in ATM Networks is the Quality of Service (QoS) guaranteed through negotiation contracts. The Quality of Service indicates the user's importance of the service, translated in terms of guaranties for delays, limits on cell losses, etc. A contribution to enabling the QoS guarantees is provided by mechanisms in the network which allow for protection against undesirable events,

like network element failures or network area overloads, threatening the continuation and quality of the service provisioning, thus degrading contracts and decreasing customer satisfaction.

Intelligent routing aims at efficiently using the available network resources with respect to survivability, congestion control and possibly support of mobility. In all of these fields there is the requirement of finding alternative routes to replace interrupted connections, whether it is a network failure, a congested link or a user movement that caused the interruption. Considering the common point of finding alternative routes, it is more logic to indicate this with the term rerouting rather than restoration, since the latter limits the application area solely to dealing with network failures. Therefor rerouting will be used categorically in the rest of the paper to emphasise the key issue in this context: to find in an efficient way an alternative route to reroute an interrupted connection.

This paper introduces the Pro-active Search protocol, which tries to take advantage of static information present in the network and known in advance. At the same time the problem of acquiring accurate real-time information is tackled using some adjusted features of distributed restoration. The paper starts with some background on rerouting and some existing rerouting possibilities in Section 2. The new rerouting protocol is presented in Section 3, describing the general principles of the mechanism, followed by a description of the phases of the protocol. The protocol performance is discussed in Section 4 and Section 5 ends with a conclusion.

## 2 REROUTING TECHNIQUES

When looking somewhat more into detail in the survivability area, various mechanisms that deal with network failures have been proposed (see below). Their main characteristics, also applicable for the general rerouting problem, are important parameters which inherently influence the performance of the protocol: the moment when the alternative route is calculated or searched, and the control under which the restoration process takes place.

With pre-planned restoration, the routes are calculated in advance taking assumptions into account on network topology, network traffic, failure scenarios etc, and they are stored in databases. Upon the occurrence of a failure the routes only have to be looked up in the database, resulting in a fast restoration technique. However this only accounts for situations that have been anticipated in advance. Unexpected situations cannot be dealt with. Restoring at real-time on the other hand, implies that the route is only searched after the failure has occurred, resulting in a slower process, but with the advantage of a more accurate technique able to react at network changes even at failure time, making it a more robust technique. An efficient balance between the two extremes could result in a robust and time efficient technique.

Considering the control of the rerouting process, there can be an overall central control centre, coordinating every action of the process. The network nodes are unable to operate without its commands, making the overall mechanism highly vulnerable. With distributed control, every network node is provided with intelligence to carry out the restoration. Therefore they exchange information and take independent decisions to obtain an alternative route. There is no central overview of the network state, the nodes search a route by exchanging requests for spare resources. A problem with this distributed control is the increased complexity, mainly due to the high volume of messages flooding the network in parallel. Another drawback is the inability to influence the process in any way and its uncertain outcome. Again these are two extremes, an appropriate balance could result in a more efficient mechanism.

Fully distributed rerouting mechanisms have already been investigated (Han Yang, 1988) (Komine, 1990) (Struyve, 1996) (Lievens, 1996), with satisfying performance results. However the above mentioned drawbacks of complexity and unpredictability have led to the research for other techniques.

The Backup Virtual Path protocol, described in (Kawamura, 1992), is such a mechanism. Basically this technique provides for every working Virtual Path (VP) a backup Virtual Path, which takes over the working traffic if the working VP is interrupted for whatever reason. The route of the backup VP is calculated in advance and the VPI/VCI translation tables of the nodes on that route are already filled in. In contrast with protection, the backup VP does not take in any bandwidth in normal working conditions. When a failure occurs in the working VP, the backup VP is activated, bandwidth is captured and the backup VP takes over the traffic of the interrupted working VP. Variations are possible, but in its basic form the issue of unexpected events especially on the routes of the Backup VPs remains: if the backup VP is disturbed, the working VP cannot be restored.

Cooperations between backup VPs and a distributed rerouting protocol, which copes with these unexpected events, have been proposed as well (Chen, 1997), however by increasing the overall rerouting complexity.

Apart from these specific efforts in the restoration area, standardisation efforts on behalf of routing, signalling and rerouting in and between private ATM networks have been ongoing, being the Private Network-Network Interface (PNNI) (ATM Forum, 1996). In PNNI networks a hierarchical routing architecture is established by recurrently dividing the network nodes in Peer Groups with parent-child relationships. All the nodes are provided with databases, describing (different) parts of the network topology. The correct operation of PNNI is based on the information on topology, routes, links etc. stored in these databases, which have to be identical for nodes within one Peer Group. A lot of effort is put in ensuring that the information in these databases is accurate and up to date, using flooding and link state based protocols. Seen the efforts for ensuring that the databases are identical and up to date, the issue was raised to develop a protocol which could



take advantage of databases with pre-stored knowledge, while at the same time trying to include accurate information without lengthy updates.

### 3 PRO-ACTIVE SEARCH PROTOCOL

In developing the rerouting protocol the following goals have been taken into account throughout the design process. A very important issue has been to end up with a protocol that performs simple and efficient rerouting. Therefore it was important to consider what can be known and thus can be done in advance, however without degrading the performance because of inaccurate information. Therefore, some real-time actions with respect to required rerouting information have to be incorporated as well.

In the following sections, the general basic ideas are discussed first, followed by a more detailed formulation of the actual algorithm phases.

#### 3.1 Background and basic principles

When considering rerouting in networks, many network aspects influence the process. Two of the most important and contributing issues are a network's topology on one hand and the network usage, i.e. the traffic or network load, on the other hand.

The topology of a network, which includes the nodes of the network and their interconnection by links, remains reasonably fixed and stable over a long period of time, certainly on the time scale important for rerouting of affected connections (i.e. seconds). The traffic in the network on the other hand, is flexible and subject to more frequent changes, as users will regularly start up new connections while other connections are torn down.

Since the topology is unlikely to change regularly on a short time basis (seconds to minutes), the topology can be regarded as a quasi-static given in the rerouting process. Therefore the rerouting protocol assumes that the network topology is known by the network nodes. This knowledge will be used to calculate candidate alternative routes in advance. Major changes in the topology, like the installation of new nodes or the upgrading of links, are distributed as updates to the nodes. This keeps them informed of the general 'static' network topology. Distributing topology information is a well-known issue and a common aspect in today's networks like for example the Internet; also in ATM PNNI routing the spreading of topology information is very important. It is emphasised that these updates only occur in an orchestrated way at regular intervals, after a major change in topology. More detailed knowledge on how these updates are performed is irrelevant with respect to the further flow of the protocol. It is noted here that an unexpected event like a network element failure is not considered as a major topological change. From this point on the overall working network topology is assumed to be known.

The traffic in the network on the contrary, is more likely to change at much shorter time intervals than the topology. This makes the network load a real-time and more uncertain aspect and it is thus considered as a dynamic issue. Therefore it is assumed that the network nodes are not aware of the current overall network traffic.

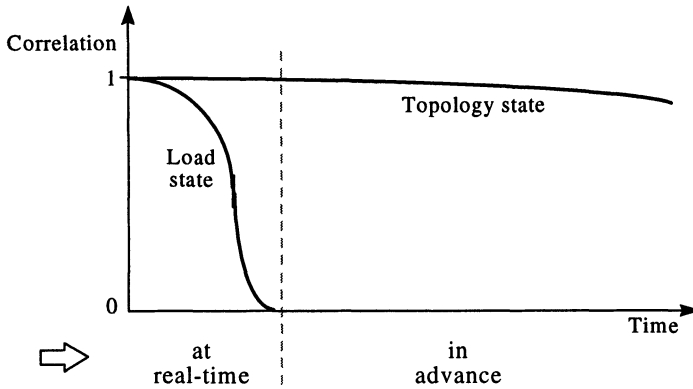


Figure 1 Network state correlation in time.

A short intuitive note on the accuracy of network information, seen from the viewpoint of the elapsed time between getting the information and actually using it, can be found in Figure 1, which shows the qualitative correlation between knowledge of the current network state and knowledge of the network state at some point in the future. The network state for the rerouting purposes described here, is divided into topology state and load state information. Given the highly static character of the topology, the topology state correlation degrades only slowly: knowledge of the current topology will most likely imply knowledge of the future topology. The network traffic is dynamic, resulting in a rapidly decreasing correlation. Knowing the network traffic now, does not imply that the traffic is known at a moment in the future, when this information might be needed for rerouting. This implies that calculations on topology can be carried out in advance, while accurate information on traffic must be collected in real-time.

When the network is confronted with unexpected changes in topology, caused for example by failures of network elements, this can interrupt working connections and it is important to restore these connections as fast as possible, in this case by finding alternative routes on which the connections can be provided again. That is dealt with by the real-time aspect of the rerouting protocol. It takes advantage of the knowledge of the network topology by using pre-calculated routes. Network traffic information is acquired at real-time using these routes. This deals with obtaining information on the bandwidth still available on links.

### 3.2 Phases in the algorithm

#### *Pre-rerouting phase: calculation of the routes*

The purpose of this pre-rerouting phase is to provide each network node with a number of paths between him and every other network node, thus taking advantage of the knowledge of the network's topology. These paths are to be used in the next phase, where the actual rerouting will take place upon the occurrence of a rerouting event.

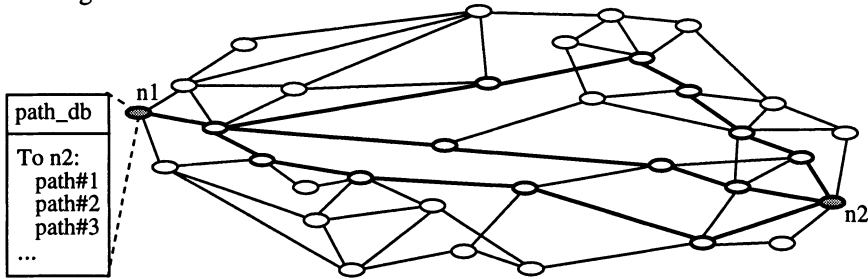


Figure 2 Pre-calculated paths.

A number of paths between every pair of nodes has to be calculated, which are stored in a database in the node (Figure 2). The paths can be determined because the topology of the network is assumed to be known to every network node. For the computation of the paths, an algorithm is required which is able to determine a number of paths between a pair of nodes, based on the topology of the network in terms of nodes and links. This algorithm can be a static and central one.

It is preferable that the routes are efficient in terms of length, cost, etc. Therefore the Pro-Active Search Protocol uses an algorithm that calculates the  $k$  shortest paths between every node pair in a network. The qualification shortest is important, since alternative routes should be efficient in terms of spare resources and since these paths will eventually be used in a time sensitive environment: the sooner an alternative route is found, the less information is lost.

Algorithms for finding  $k$  shortest paths between two nodes in a network can be found in literature. The algorithm used here is the algorithm of Yen, which finds  $k$  shortest loop-less paths in a network. The full description can be found in (Yen, 1971). It uses an arbitrary iteratively applied shortest path algorithm, in this case Dijkstra's. Furthermore Yen's algorithm avoids loops in the paths, basically avoiding that nodes are present in a path more than once. This feature is important for the efficiency of the rerouting. As far as computation time is considered, Yen's algorithm has an upper bound which changes only linearly with the number  $k$ .

It is however important to note that basically any algorithm can be used here, which is able to calculate  $k$  shortest paths from a network topology in which the nodes, the links and their costs are given. A short overview is given in (Shier, 1997).

In moderate sized networks it is feasible to calculate  $k$  paths between every pair of two nodes in the network, in other words to have every node store  $k$  paths to every other network node. When networks grow larger in size, this might become unpractical. An option is then to limit the pairs in geographical distance and only calculate paths between any pair of nodes not too far away from each other. Another solution can be found in hierarchical networks, like for example in PNNI networks [7]. There the paths can be confined within a peer group, limited in size anyway, and calculated between any pair of peer nodes.

It is not considered efficient to store all the paths of the entire network or network group in some central location, which is consulted after a failure and which downloads the required paths. This is considered to be too vulnerable and time consuming. In stead every node has its own database, in which paths from this node to all or some other nodes are stored. When required, the paths are immediately available.

The paths have to be computed prior to a rerouting event, explaining the term pro-active. Whenever a major topology update is issued, the routes must be recalculated, to ensure the effectiveness of the paths. This recalculation - using the same  $k$  shortest path algorithm as before, but now on the updated topology - is a background process, running in parallel or more accurately in between rerouting processes. Possible inconsistencies of databases between nodes are here not as important or threatening as for example in PNNI. Because a node knows the entire path, it must not rely on a next intermediate node to determine the appropriate next hop in the path. The network nodes don't need identical databases to ensure a correct result. Moreover, since a number of paths are pre-calculated, the impact of one invalid route is low, at most only slightly decreasing the performance.

### *Rerouting phase*

This is the part of the rerouting protocol that executes real-time actions in order to collect real-time link information. This will result in obtaining valid alternative routes, which will actually replace the failed connections. The required real-time information on the load of the network is collected by sending out search messages - so-called Courier messages - in a controlled way.

This real-time rerouting part is triggered by a so-called rerouting trigger event. This is an unexpected network event, like the failure of a network element, interrupting working connections. Also a network link becoming overloaded with risk for congestion, can act as a trigger to search alternative routes avoiding this crowded link.

As far as the rerouting is concerned, there is the choice to reroute on link or on path basis (Figure 3). With link rerouting, only the unavailable link will be replaced by an alternative route, the undamaged parts of the connections using the failed link are kept. In case of path rerouting, the entire connection(s) using the failed link will be rerouted. The latter needs some extra complexity because the connection endnodes must be notified of the rerouting and the capacity along that

connection on working links must be released. This requires some extra delay as compared with link rerouting, which is a local process and generally faster. However the total alternative route in link rerouting might be less efficient.

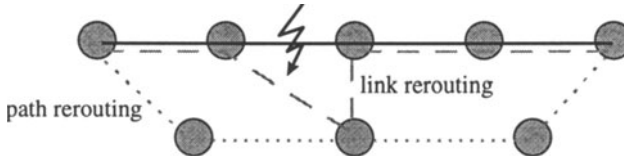


Figure 3 Link and Path rerouting.

The nodes between which the alternative route is searched - whether it are the link adjacent nodes or the connection end nodes - are called the restoration node pair. They will start up and eventually terminate the real-time process. In this protocol, the Sender-Chooser technique (Grover, 1987) is used, in which one of the restoration pair nodes starts sending out Rerouting Messages, while the other one will eventually choose the alternative routes found during the rerouting process. They are called Sender and Chooser node (Figure 4). The arbitration of Sender and Chooser is mostly based on node IDs. It should be noted that in case of path rerouting different Sender-Chooser pairs will be active in the network, one for each failed connection. This can complicate the process.

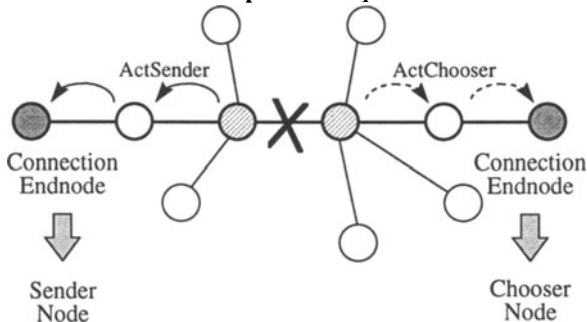


Figure 4 Sender-Chooser arbitration.

### Actions of Sender and Chooser nodes

The Sender node consults its database for the  $k$  pre-calculated shortest paths to the Chooser node. On each of these  $k$  paths, the Sender node sends one Courier Message to the first node on these routes, which represent the possible alternative routes (Figure 5). Every Courier Message collects and stores real-time information about that route on its way to the Chooser node. Basically this deals with the available bandwidth of the links on the route.

The Chooser node will initially just wait, until Courier Messages arrive from any of the routes.

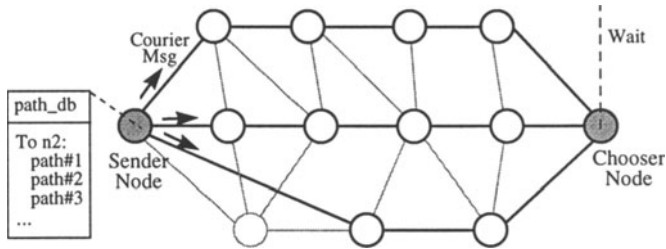


Figure 5 Actions of Sender and Chooser.

### Controlled flooding: forwarding of Courier Messages

The Courier Messages are forwarded from node to node along their specific route, meanwhile collecting appropriate real-time link information (Figure 6).

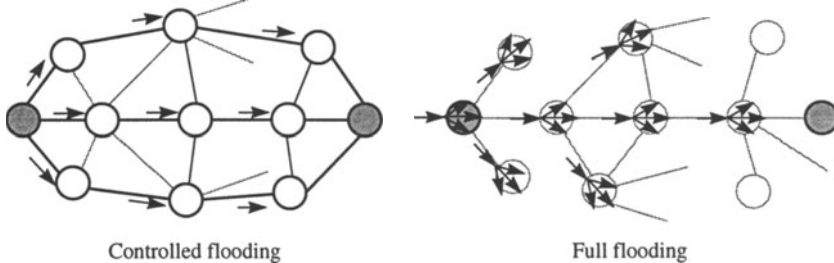


Figure 6 Controlled vs. full flooding.

This phase has characteristics of a distributed restoration algorithm: the network nodes perform the required rerouting actions without commands from some central control unit. Some of the major drawbacks of fully distributed algorithms however - being the uncontrolled flooding of restoration messages, the resulting complexity and the unpredictable outcome of the restoration process - is avoided by the controlled forwarding of Courier Messages. The messages are not flooded to all neighbours, which could create an avalanche of parallel messages roaming in the network, but they are only sent on specific and fully specified paths. Also the network area extent which is visited by the messages is controlled: only the links of the pre-calculated routes will carry rerouting messages. This implies that the network operator has more control on the alternative routes to be found, while still having the advantage of self-healing facilities in the nodes.

The rerouting messages do not reserve bandwidth along the route. They only collect the real-time load information and bring this information to the Chooser node. This strategy avoids complex release protocols at the end of the process, required to release bandwidth previously reserved for a particular failure but which is no longer needed because, for example, other routes have been found.

A Courier Message will only be forwarded on the next link of the route if there is still some available bandwidth on that link. This implies that a Courier Message encountering a link without spare capacity is not forwarded (Figure 7). As far as

that particular route is concerned, the rerouting process stops: this is not a viable alternative route. No cancel or release messages are required.

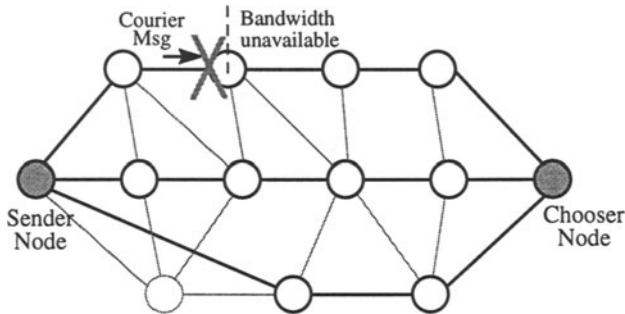


Figure 7 Pre-calculated path unavailable.

### Actions of the Chooser node

Assume that a Courier Message arrives in the Chooser node. With the collected information about the links' load in the message, the Chooser node can build up an overview of used and available capacity on the routes between Sender and Chooser. On that basis the Chooser can observe whether a route has available capacity and decide to take a route as actual alternative route. The Chooser consequently adjusts its capacity allocation overview to indicate that some of the available capacity is taken by an alternative path. This is mainly important when the  $k$  routes have some links in common and conflicting situations in the allocation of the available capacity can occur. The choice of an actual path by the Chooser is done on first-come first-take basis. It is also possible to wait for a certain time period and then to choose between the arrived candidates the route with the largest sufficient bandwidth.

### Route confirmation

When a route is chosen, the Chooser node sends back a Confirmation Message on that route towards the Sender node. This message will ensure that the nodes on the route will adjust their routing tables and actually reserve the capacity for that route. In that way, the message is rippled back to the Sender node, which then knows that a valid alternative route is found.

It is possible that on the way from Chooser to Sender, a link is encountered where the capacity assumed to be free is no longer available because for example another rerouting process has taken in that bandwidth. In that case a Cancel message is sent back to the Chooser node and, if available, another route is taken in stead.

The rerouting process is terminated when all failed capacity can be rerouted along alternative route(s) or when there is no more available bandwidth on the  $k$  routes. In the latter case some of the failed capacity cannot be rerouted.

### *Remark on the $k$ shortest paths*

As can be derived from the previous sections, the nature of the pre-calculated paths highly contributes to the success chances of the protocol. At the end of the rerouting phase, some of these paths will be used as the actual alternative route(s) for the affected connection. It is obvious that these paths will seriously affect the final rerouting performance.

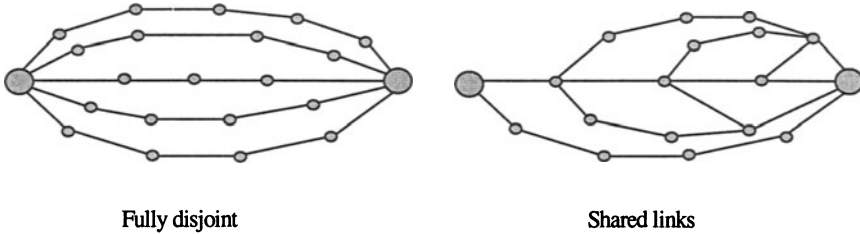


Figure 8 Nature of the pre-calculated paths.

The number of pre-calculated paths is an important parameter in the rerouting process, influencing the performance and the rerouting capabilities of the protocol. The number  $k$  must be large enough to have reasonable chances that a feasible route can be found eventually.

The paths must also be efficient in terms of the used links. This is concerned with how the  $k$  calculated paths between a node pair are related to each other, with respect to using the same links and nodes (Figure 8). With Yen's algorithm it is highly probable that the  $k$  shortest paths between two nodes share a number of links and nodes. If such a shared link has little available capacity, this means that immediately a number of the  $k$  paths become unavailable as alternative routes, degrading the rerouting chances. The number  $k$  does not directly refer to the number of disjoint paths between a node pair. However it is important to realise that fully disjoint paths are not required for this protocol, simply a number of pre-calculated routes covering a certain area which can be searched in a controlled way at real-time. Besides, the connectivity degree of the network limits the maximum number of disjoint paths due to the limited number of outgoing links out of a node.

An simple extension has been added to the Yen algorithm to avoid that in an extreme case a link is used by all of the  $k$  pre-calculated paths. Basically a threshold is used to set the maximum number of paths that can use a same link. This increases the rate of success, since this one link is a possible bottleneck to the rerouting process. The protocol leaves ample space for other algorithms that can calculate  $k$  paths between two nodes.

The use of pre-calculated paths also allows for a more particular choice of the routes. In stead of just calculating the  $k$  shortest paths, one could provide the databases with paths that cover only a certain network area, recommended for rerouting. This would allow network operators to enforce a certain rerouting strategy.



#### 4 PROTOCOL PERFORMANCE

In order to verify the behaviour of the protocol as well as to assess its performance in terms of rerouting capacity, the protocol was described in the standard language for protocol description, being the Specification and Description Language SDL (Z.100, 1988), by using the SDL Design Tool SDT, which provides the means to describe and simulate communication protocols.

Each node is assigned a model of a FIFO queue and one processor. It takes 10 ms to read in and process an incoming message and 10 ms to generate an outgoing message. The simulations were performed on a realistic 32 node network, provided with enough spare capacity to enable the rerouting of all single link failures.

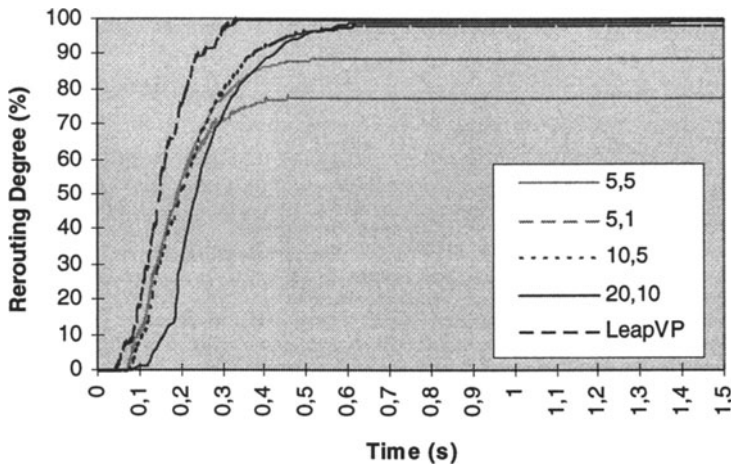


Figure 9 Single link failures (k, threshold), link rerouting.

The average rerouting degree for link rerouting of all single link failures in the sample network as a function of the elapsed rerouting time in Figure 9. When  $k$  is small, there is usually not enough total spare capacity to reroute all failed connections. The combination (20,10) showed the best degree. When comparing with a distributed technique simulated with the same parameters (Lievens,1996), it shows that indeed the in advance actions of this protocol do give good results.

The same network was simulated for single link failures applying path rerouting. The obtained results are significantly less than with link rerouting. This is due to the fact that with path rerouting, different simultaneous search processes are ongoing in the network, one for each connection that used the failed link. These simultaneous processes compete with each other for spare capacity, thus explaining the performance degradation.

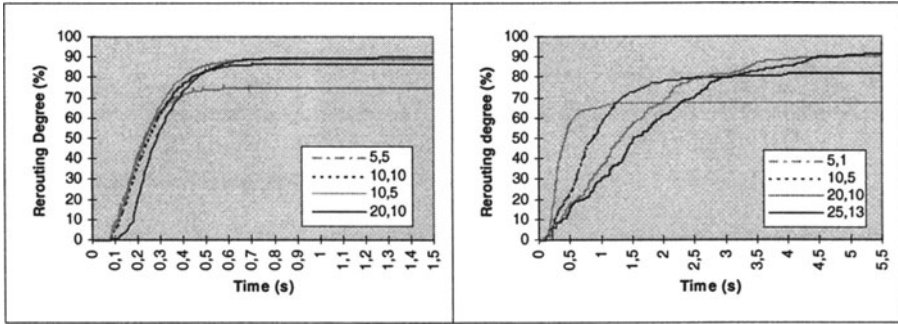


Figure 10 Double link failures; node failures.

Simulations were performed for node failures and double link failures (Figure 10), on the same network with fixed spare capacity, resulting in satisfying performance. For the node failures, path rerouting was used, explaining the longer time needed to achieve rerouting.

## 5 CONCLUSION

In this paper a new rerouting protocol is introduced for ATM networks. The Pro-Active Search protocol combines real-time action with path calculations carried out in advance. This is based on the assumption that a network's topology is quasi-static. Therefore this topology is assumed to be known by the nodes, and they take advantage of this by calculating shortest paths between them and every other network node. The result is that each node knows  $k$  paths to another node, as far as the general topology is concerned. The network's traffic is assumed to be dynamic and flexible; this is considered as unknown by the nodes. When rerouting is required, the nodes will gather accurate, real-time information of the load on the pre-calculated paths by sending a limited number of Courier messages on each of the  $k$  paths. In this way speed and accuracy are combined, resulting in a fairly simple protocol with the advantages of pre-planned restoration together with those of real-time distributed restoration.

### *Acknowledgments*

This work has been supported by the Flemish Government through the IWT project ITA/950214/INTEC.

## 6 REFERENCES

- Han Yang, C. and Hasegawa, S. (1988) FITNESS: Failure Immunization Technology for Network Survivability. *Globecom '88*, 1549-54.
- Komine, H., Chujo, T., Ogura, T., Miyazaki, K. and Soejima, T. (1990) A Distributed Restoration Algorithm for Multiple-link and Node Failures of Transport networks. *IEEE 1990*, 459-63.
- Struyve, K., Demeester, P., Nederlof, L. and Van Hauwermeiren, L. (1996) Design and Evaluation of distributed link and path restoration algorithms for ATM meshed networks. *Proceedings International Zurich Seminar on Digital Communications, ETH Zurich, Switzerland*, Vol. 1044.
- Lievens, I. and Demeester, P. (1996) The use of distributed restoration in intelligent routing. *Proceedings of IEEE Fourth Symposium on Communications and Vehicular Technology in the Benelux*.
- Kawamura, R., Sato, K. and Tokizawa, I. (1992) Self-healing ATM Networks Based on Virtual Path Concept. *Networks*, 129-34.
- Chen, S. and al. (1997) An Integrated Restoration Approach (IRA) in the ATM Network. *Globecom 1997*, 1388-92.
- The ATM Forum Technical Committee. (1996) Private Network-Network Interface Specification Version 1.0 (PNNI 1.0). *af-pnni-0055.000*, March 1996.
- Yen, J.Y. (1971) Finding the k shortest loopless paths in a network. *Management Science*, Vol. 17, 712-16.
- Shier, D.R. (1979) On Algorithms for Finding the k Shortest Paths in a Network. *Networks*, Vol. 9, 195-214.
- Grover, W.D. (1987) The Self-Healing Network: a fast distributed restoration technique for networks using digital cross-connect machines. *Globecom 1987*, 1090-5.
- ITU Recommendation Z.100 SDL-88. (1989) Functional Specification and Description Language.

## 7 BIOGRAPHY

Ilse Lievens graduated in Electrical engineering at the University of Gent in 1994. Her graduation thesis ("The influence of restoration strategies on the performance of wavelength multiplexed networks") dealt with static and centralised restoration in WDM networks. Since September 1994 she is working in the Broadband Communications Networks Group as a research associate and is preparing a Ph.D. Her research interests include survivability issues in ATM networks. In this context she was involved in the RACE II project IMMUNE, and the set-up of the PANEL project. Her research is extended towards intelligent routing, focusing at a common rerouting strategy for survivability, congestion and mobility.

# Impact of VC Merging on Buffer Requirements in ATM Networks

*A.L. Schmid\**

*Swiss Federal Institute of Technology, ETHZ  
8092 Zurich, Switzerland*

*e-mail: Andreas.Schmid@switzerland.org*

*\*Work performed at IBM Research Division, Zurich Research Laboratory.*

*I. Iliadis and P. Droz*

*IBM Research Division, Zurich Research Laboratory  
8803 Rüschlikon, Switzerland*

*Tel. +41-1-724-86-46, Fax +41-1-724-89-55*

*e-mail: ili,dro@zurich.ibm.com*

## Abstract

For the implementation of multipoint-to-point connections in ATM, various approaches exist, each with its own advantages and disadvantages. VP-based methods require unique sender identification but they do not require reassembly in merging points. In contrast, VC-based methods do not require unique sender identification but they do require reassembly in merging points. It is likely that VC merging will be the method of choice as it is scalable and yet relatively simple to implement. One of its drawbacks is the increased output buffer space required at the switches because of packet reassembly at the merging points. This paper investigates the impact of the switch architecture and characteristics on the output buffer space by means of simulation. The results obtained demonstrate that for typical switch architectures, VC merging does not require significant additional buffering compared to VP merging.

## Keywords

IP over ATM, VC merging, VP merging

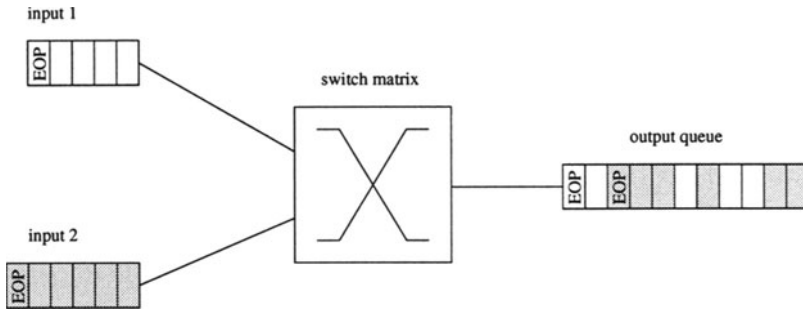
## 1 INTRODUCTION

In current ATM networks, there exist only *point-to-point* (pt-to-pt) and *point-to-multipoint* (pt-to-mpt) connections. For the interconnection of routers across an ATM network as well as for many other information-gathering applications, *multipoint-to-point* (mpt-to-pt) connections appear to be more appropriate. Interconnection of  $N$  routers requires order  $N^2$  labels for the order of  $N^2$

pt-to-pt connections as described by Calvinac *et al.* (1997). With mpt-to-pt connections only  $N$  labels for the  $N$  associated connections are necessary. This significantly reduces the required label space and thus makes the method more scalable. The same new ATM connection type could also be used in the context of merged connections for MPLS (Callon *et al.* 1997).

To implement mpt-to-pt connections, different solutions are possible. We focus on the two most important methods: *VP merging* and *VC merging*.

*VP merging*: Each sender is assigned a globally unique identifier having the format of a VCI. The identifier is carried in the VCI field of the ATM cell. The ATM switch translates incoming VPIs for the same destination to the same outgoing VPI. The receiver distinguishes amongst the different sources by the different VCIs. The key advantage of this scheme is that no VCC resources are required in the switching nodes as only VP switching is performed. This implies no change of hardware but only a change of the connection establishment protocol. Some of the disadvantages of VP merging are the lack of scalability caused by the VPI address space limitation of 4096 entries and the need for a "global VCI uniqueness" protocol. There are proposals to circumvent the nonscalability by enlarging the VPI address space at the expense of VCI address space. This is not desirable, however, because it requires changes in the switching hardware. *VC merging*: This method avoids the requirement for globally unique sender identifiers, and it consumes only one VCI per traversed link. These characteristics make this approach scalable. Each source participating in a mpt-to-pt connection has a unique VCI per link. The ATM switch translates incoming VCIs belonging to the same connection to a single outgoing VCI. This means that cells of packets belonging to different senders could be interleaved. As the receiver is not able to distinguish cells from different senders, packet reassembly has to be performed at the merging points, and all cells from a given packet must be sent contiguously so that reassembly at subsequent merging points and at the receiver will be possible. AAL 3/4 would solve the problem by introducing the Message Identifier (MID) field for sender identification in every cell. The use of AAL 3/4, however, has other drawbacks such as the limited space of the MID field, the inefficient encapsulation method, and the less powerful CRC capability. In this paper we consider the employment of AAL 5 because it is widely available and supported in ATM switches, especially in data networks. Packet reassembly at the merging points introduces additional buffer requirements on the switching architecture because all of the cells of a packet sent by a sender belonging to a mpt-to-pt connection have to be stored and must wait for the last cell of the packet identified by the "End Of Packet" (EOP) marker used by AAL 5 to arrive. Figure 1 depicts the cell interleaving problem. Packet reassembly also introduces additional delay for packets transported over a merged connection and adds burstiness to the traffic. This is because all the cells of a packet have to wait at every merging point. They appear afterwards as a burst of a whole packet at the output link. This burstiness becomes even worse



**Figure 1** Cell interleaving problem.

as it is often cascaded and thus accumulated over numerous merging points. Heinanen (1997) gives some hints about how to solve the problems involved in mpt-to-pt VC merging.

A third possibility is to handle a mpt-to-pt connection of  $N$  senders to one receiver like  $N$  pt-to-pt connections without applying any merging. This possibility again requires order  $N^2$  labels for the order of  $N^2$  pt-to-pt connections. Of the above possible solutions, VC merging appears to be the method of choice as it is relatively easy to implement and yet scalable. At the ATM Forum, VC merging has been almost fully accepted and will most likely be introduced in the PNNI v2.0 specification (expected to be finished in the spring of 1998). The only concern is with the reassembly required in the switches in terms of additional buffering and delay. The numerous simulations presented in the following sections are used to investigate the required additional buffer overhead for VC merging. It is also very likely that different methods of merging and nonmerging will exist simultaneously in an ATM network. Some interworking aspects of these methods are discussed by Widjaja *et al.* (1997).

Section 2 of this paper describes our switching architecture model for VC merging and the model of the arriving traffic. In Section 3 we show our simulation setup and discuss the results of the simulations. In Section 4 we give a summary and derive some conclusions.

## 2 SWITCH AND TRAFFIC MODEL

### 2.1 Switch Model

In this paper we consider the general class of single-stage, nonblocking  $M \times M$  packet switches with both input and output queuing (Iliadis and Denzel 1993, Denzel *et al.* 1995). The shared output buffer is assumed to be sufficiently large so that the switch performance is close to optimal, corresponding to the pure output queuing. Cells are transferred from the head of the input queues to the shared buffer. The speed of the input and output switch ports is denoted  $R_S$ ,

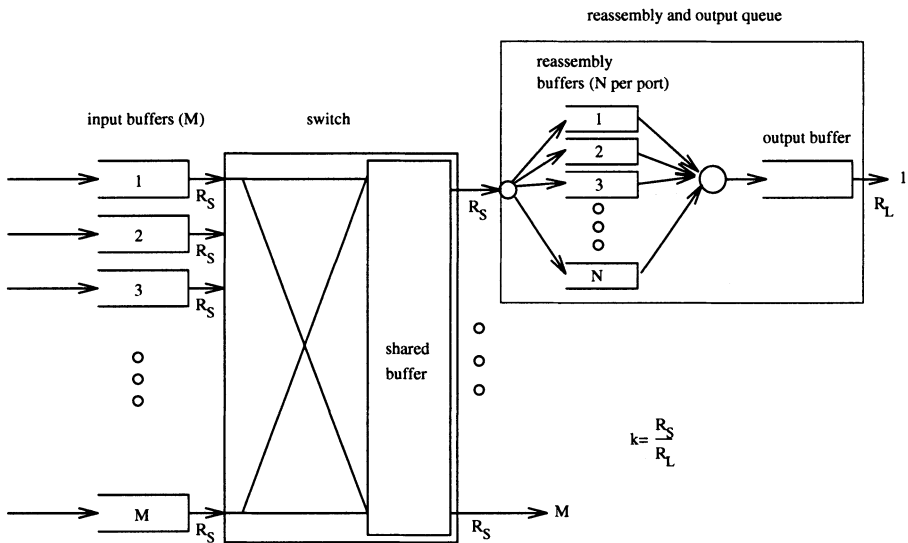
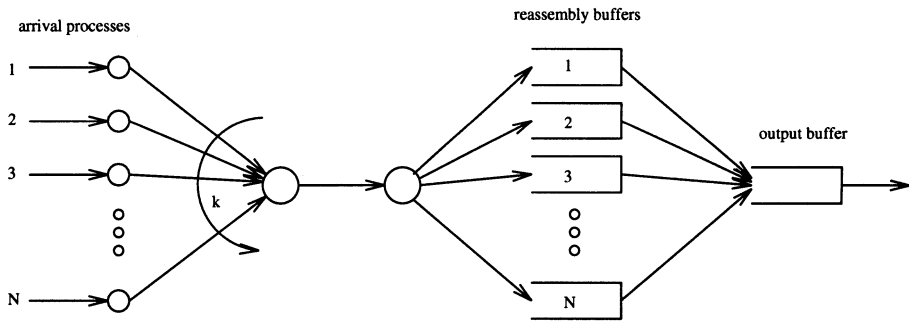


Figure 2 Switch architecture.

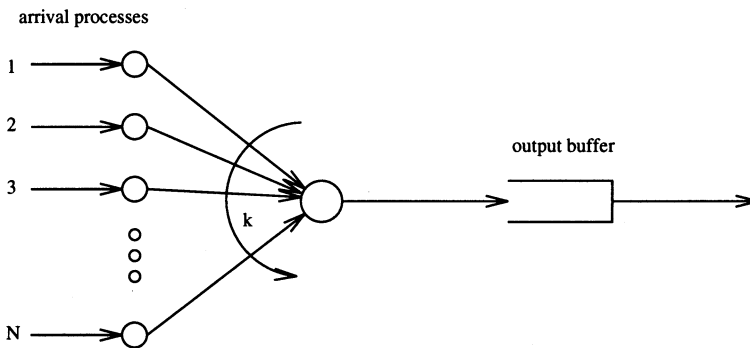
and the speed of the outgoing links is denoted  $R_L$ . Let  $k$  denote the speed ratio of the switch speed (per port) to the outgoing link speed, i.e.  $k = R_S/R_L$ . Typically  $k$  is greater than one, which implies that an output queue should be provided in order to cope with the speed mismatch.

As described above, VC merging requires an amount of additional output buffering due to the packet reassembly. We introduce a so-called reassembly buffer at each output port of the ATM switch. Figure 2 shows the concept of the reassembly buffer. A switch has  $M$  input ports and  $N$  sources of mpt-to-pt connections because it is likely that different connections will coexist. Hence  $N$  can be much larger than  $M$ . The model considered in this paper is valid for the case of  $N \leq M$ . The case of  $N > M$  is not covered by the present switch model and is therefore a subject for further investigation. At every merging point, each of the sources participating in the corresponding mpt-to-pt connection is associated with a distinct reassembly buffer at the output queue. When the last cell of a packet with the EOP marker arrives at the reassembly buffer, all of the cells of a packet are instantly transferred into a single output buffer per output port. Physically the reassembly and the output buffers of one output port share a common memory pool. The transfer from the reassembly to the output buffer can easily be done by a pointer movement and will therefore not incur additional delay.

The simulation models for VC and VP merging are shown in Figures 3 and 4, respectively. Cells belonging to the various VCs are transferred from the head of the switch input queues in the shared buffer and, subsequently, to the corresponding output queues. It is assumed that the traffic is uniform, i.e. the destination of an arbitrary packet can, with an equal probability, be any



**Figure 3** Our simulation model for VC merging.



**Figure 4** Our simulation model for VP merging.

of the output ports, and that successive packets are independent regarding their output port destinations. Owing to traffic symmetry, all of the output queues have identical behavior. Let us turn our attention to a particular output queue and study its behavior. The corresponding simulation model considers  $N$  sources feeding the output queue in a round-robin fashion governed by the factor  $k$ . This model is also appropriate for the case where the switch fabric is capable of transferring only a limited number of cells to any given output (Oie *et al.* 1989).

## 2.2 Traffic Model

The traffic and simulation model we use is shown in Figures 3 and 4. We use  $N$  arrival processes, which correspond to the traffic destined to the output queue. Packets are assumed to arrive according to either a Poisson process (nonbursty traffic with the mean arrival rate  $\lambda$ ) or a hyperexponential process (bursty traffic). The hyperexponential process is generated by a two-stage hyperexponential distribution. The mean values corresponding to the two stages



are  $0.51 * \lambda$  and  $16.48 * \lambda$ , respectively. The corresponding routing probabilities for the two stages are 0.97 and 0.03, respectively, so that the mean arrival rate is again equal to  $\lambda$ . Each packet is assumed to contain a number of cells geometrically distributed with a mean of  $E$  cells (Chao and Smith 1992, Widjaja and Elwalid 1997). We used  $E = 10$ ,  $E = 30$ , or  $E = 180$  cells (10 cells correspond to 472 bytes, 30 cells to 1432 bytes and 180 cells to 8632 bytes).

It is shown by Widjaja and Elwalid (1997) that the mean packet size in a core network where ATM is likely to be applied is about 289 bytes. This yields 6.2 ATM cells of data using AAL 5 with the null encapsulation method as described by Heinanen (1993) (additional overhead of AAL 5 is 8 bytes per packet). The dominant packet sizes in an Internet backbone are 40 or 44 bytes at about 36% of the traffic (TCP acknowledgment packets, TCP control segments such as SYN, FIN, ..., and Telnet packets carrying single characters), 552 or 576 bytes at about 25% (512 and 536 bytes of TCP implementations without path MTU discovery as the default maximum segment size (MSS) for nonlocal IP destinations, yielding a 552 or 576-byte packet size), 185 bytes at about 2.7%, and 1500 bytes at about 1.5% (Ethernet traffic). These statistics were collected on Feb 10, 1996, in FIX-West network as a sample wide-area network, and are given on the NLANR homepage (1996). A more recent study of traffic characteristics in an Internet backbone was conducted in August of 1997 (Thompson *et al.* 1997). It is shown that almost 50% of the traffic is 40 or 44 bytes in packet length. More prominent packet sizes are 532, 576, and 1500 bytes, each representing 15% of the traffic. Comparing the two studies we observe a shift to smaller packets of size 40 or 44 bytes and larger packets of size 1500 bytes.

For the future development of packet sizes, the spreading of the use of path MTU (PMTU) discovery will have a significant impact. PMTU will affect MTUs in IPv4 as proposed by Mogul and Deering (1990) and even more MTUs in IPv6 over faster LANs. There will be numerous packets with possible sizes up to 64 kilobytes (max. packet format for AAL 5 is 64 kilobytes (Laubach 1994)). A single packet of this size involved in reassembly could alone fill the entire reassembly buffer in a switch output queue. Atkinson (1994) gives an overview of other typical frame sizes being applied on AAL 5. These are 8 kilobytes used by the Network File System (NFS) and the 9180 bytes of IP MTU over SMDS (Piscitello and Lawrence 1991) that became the default value for IP MTU over ATM AAL 5 (Laubach 1994). These big packet sizes in conjunction with VC merging could induce present problems that VP merging would not encounter. On the other hand there will also be much more real-time traffic (e.g. voice) in the Internet. Real-time traffic typically produces a large amount of very small packets.

### 3 SIMULATION SETUP AND RESULTS

This study concentrates on the additional buffer space required for reassembly. It is conducted under loads  $l = 30\%, 70\%, 90\%$ , with different traffic characteristics (bursty, nonbursty), factors  $k = [1, \dots, 16]$ ,  $N = 16, 64, 128$  sources,  $M$  ports ( $M \geq N$ ), and with a mean packet size of  $E = 10, 30, 180$  cells. The default values for the simulations are  $N = 16$  sources,  $E = 10$  cells,  $l = 90\%$ , and  $k = 16$  unless specified otherwise. Figures 5-13 show the VC merging buffer size (solid line) and the corresponding VP merging buffer size (dashed line). The results serve to compare VP and VC merging. They cannot, however, be used directly to show the required output buffer space in an ATM switch because no flow control has been taken into account. The simulations were carried out for an extremely large number of events such that 95% confidence intervals were very small.

Figure 5 shows the results for  $l = 30\%, 70\%, 90\%$ . The difference between the solid and the dashed lines (VC and VP merging for a specific load) is about 19 to 21 cells for  $l = 90\%$ , about 21 to 23 cells for  $l = 70\%$ , and about 30 to 37 cells for  $l = 30\%$  over some magnitudes of overflow probability. At high loads the output queue contains a large number of cells, which translates to long delays. Therefore, by the time the first cell of a packet is ready for transmission at the output link, the corresponding last cell has most likely arrived and, consequently, the packet reassembly has been completed. In this case, therefore, the additional overhead due to reassembly is almost negligible. It is also important to note that the workload of today's switches normally lies at high levels of around 70% or 80%. In contrast, at low loads, the first cell may be ready for transmission while the reassembly is in progress. In this case it has to be delayed until the reassembly process has been completed. How-

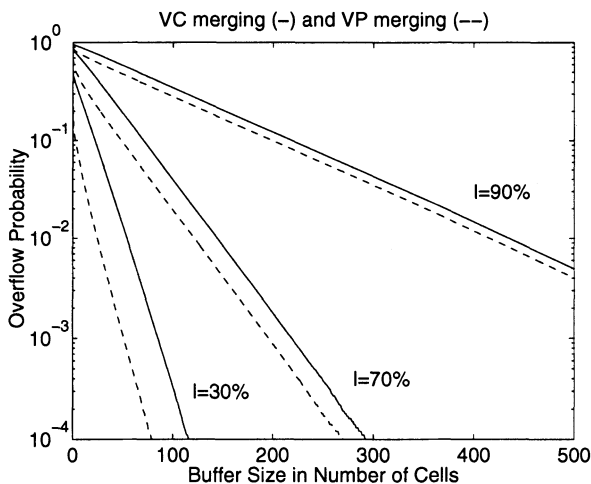
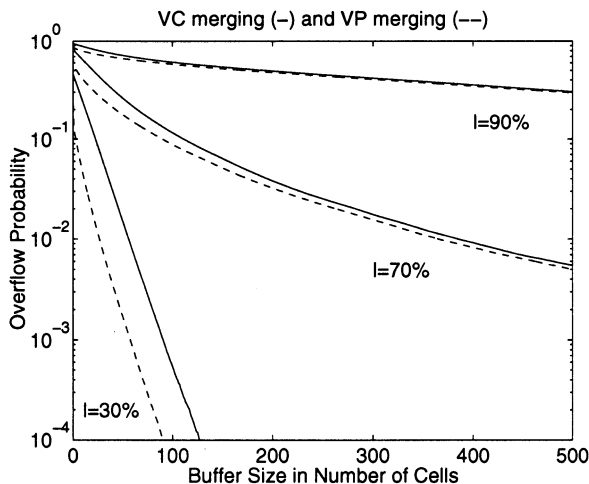


Figure 5 Simulations for  $l = 30\%, 70\%, 90\%$ , nonbursty arrival process.

ever, owing to the low load, the number of packets under reassembly is small and, therefore, the additional buffer requirement of VC merging is minimal. The results obtained are in agreement with those presented by Widjaja and Elwalid (1997). Furthermore, the packet delay corresponding to VC merging approaches that corresponding to VP merging as the load increases.

We then made the same simulations with bursty arrival processes. We model the bursty arrival process by a hyperexponential packet arrival process as described in the previous section. The results for  $l = 30\%$ ,  $70\%$ ,  $90\%$  are shown in Figure 6. We see that the buffer requirements for both VC and VP merging grow significantly for high loads. Of course flow control would alleviate this problem to some extent due to the overall load reduction. The additional buffer requirements for VC merging compared to VP merging are minimal even for the case of bursty traffic. In particular, for high loads they become negligible for the reasons given above.

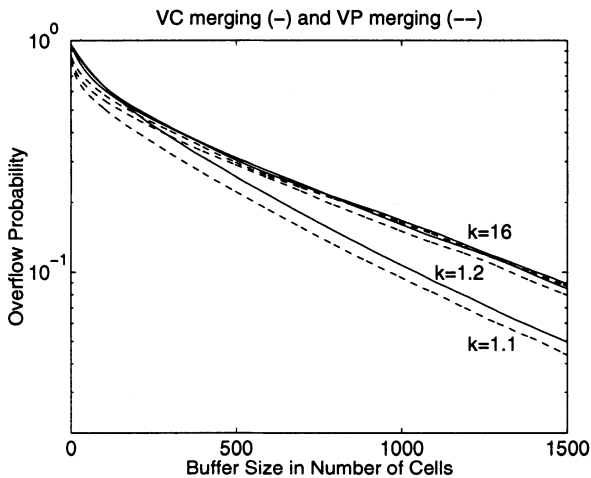
Simulation results were obtained for different values of  $k$  and different loads  $l$ . By varying  $k$  we expected to see an influence on the additional buffer requirement. Surprisingly, only the extreme value  $k = 1$  resulted in a big additional buffer requirement for VC merging. It is obvious that VP merging requires almost no output buffer with  $k = 1$  as the speed of the switch output port ( $R_S$ ) is equal to the speed of the output link ( $R_L$ ). We then tried to determine the critical  $k$  for every load factor  $l$  considered. The critical value of  $k$  is defined as follows: For all values of  $k$  larger than the critical value, there is practically no distinction between VC curves and VP curves, whereas for all smaller values of  $k$  the curves start becoming distinguishable. We found that the critical  $k$  lies close to the extreme value  $k = 1$ . The range of the critical  $k$  is between 1.1 and 1.3 for  $l = 90\%$  and  $l = 70\%$ , respectively. This means that the critical  $k$  becomes larger with lower loads, but it is still far away from



**Figure 6** Simulations for  $l = 30\%$ ,  $70\%$ ,  $90\%$ , bursty arrival process.

the values implemented in today's switches (greater than 2). To substantiate these observations we investigated the critical  $k$  for  $l = 30\%$ , too. In this case the critical value for  $k$  is approximately 1.5, which is still much smaller than 2 and thus confirms our theory.

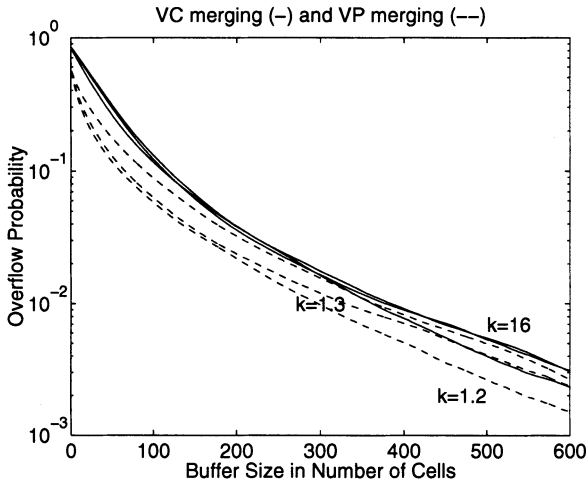
Figure 7 shows the results of our simulations for  $k = 1.1, 1.2, 16$  at  $l = 90\%$ . We observe that all of the curves for the output buffer of VC merging at different values of  $k$  lie close together. The value of  $k = 1.1$  is the critical one because the corresponding curve starts to show a deviation. The same applies to the curves for VP merging. For values of  $k$  greater than the critical one, the additional buffer requirement for VC merging at low overflow probabilities is minimal. However, the difference between VC and VP merging becomes noticeable for values of  $k$  less than the critical value.



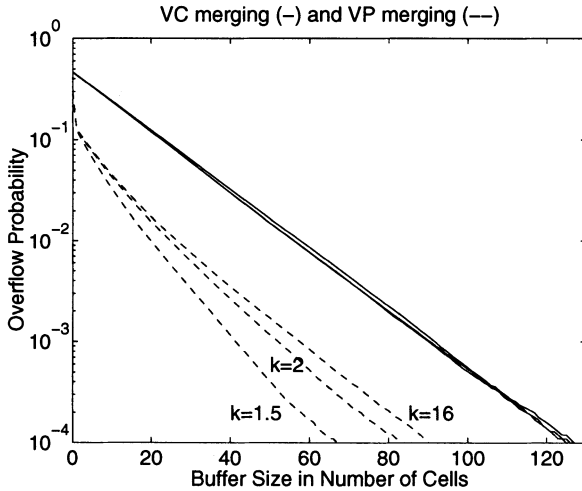
**Figure 7** Simulations for  $l = 90\%$ ,  $k = 1.1, 1.2, 16$ , bursty arrival process.

Figure 8 shows the results of our simulations for  $k = 1.2, 1.3, 16$  at a lower load of  $l = 70\%$ . In this case, we observe that the different curves for VC and VP merging lie fairly close together with the exception of the curves for  $k = 1.2$ . This shows that the critical  $k$  is slightly larger for  $l = 70\%$  (about  $k = 1.2$ ) than for  $l = 90\%$  (about  $k = 1.1$ ). Here again, the additional buffer requirements for VC merging at low overflow probabilities become noticeable for values of  $k$  less than the critical value.

Figure 9 shows the results of our simulations for  $k = 1.5, 2, 16$  at a low load of  $l = 30\%$ . We observe again the similarity of the curves for VC merging over the entire range of  $k$ . The curves for VP merging vary slightly so that the additional buffer space becomes smaller for a larger  $k$ , with a critical  $k$  at about  $k = 1.5$ . There is a noticeable additional buffer requirement for



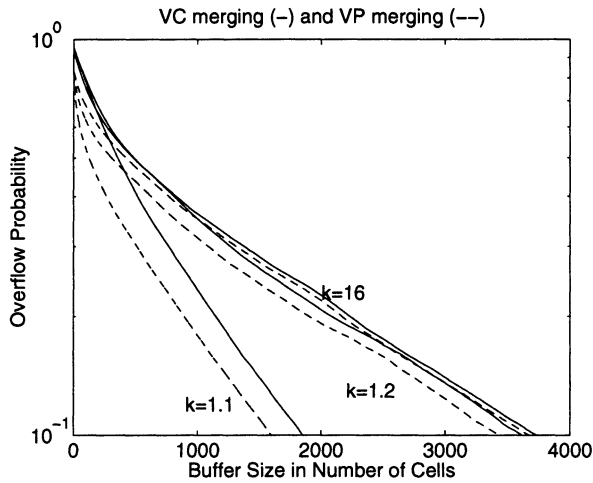
**Figure 8** Simulations with  $l = 70\%$ ,  $k = 1.2, 1.3, 16$ , bursty arrival process.



**Figure 9** Simulations for  $l = 30\%$ ,  $k = 1.5, 2, 16$ , bursty arrival process.

VC merging in the entire range of values of  $k$ . Furthermore, the additional requirement increases as  $k$  decreases.

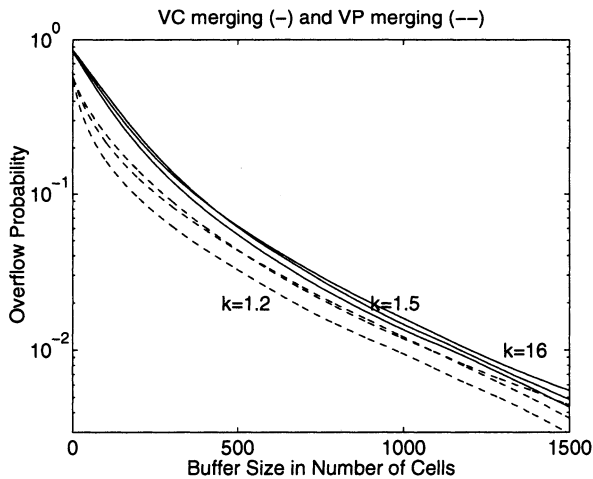
We then tried to investigate the possible influence of more specific traffic characteristics such as larger packet sizes and increased numbers of sources in a mpt-to-pt connection on the additional requirements of VC merging compared to VP merging. First, we performed simulations for a larger mean packet size of the arrival process ( $E = 30$ ). Figure 10 shows the curves for  $l = 90\%$  and  $k = 1.1, 1.2, 16$  with a mean packet size of  $E = 30$ . Compared to Figure 7 we observe a greater difference between the curves for  $k = 1.1$  and for  $k = 1.2$ .



**Figure 10** Simulations for  $l = 90\%$ ,  $k = 1.1, 1.2, 16$ , bursty arrival process and  $E = 30$ .

It appears that the critical  $k$  is shifted to a value slightly larger than  $k = 1.1$  (between  $k = 1.1$  and  $k = 1.2$ ). Furthermore we see that the mean packet size, which is three times larger, requires an output buffer size that is also three times larger. Moreover, the additional output buffers for VC merging are about three times larger for  $E = 30$ . Therefore the additional buffer requirement for VC merging appears to grow linearly with the mean packet size. This trend is also verified by our simulations for  $E = 180$ .

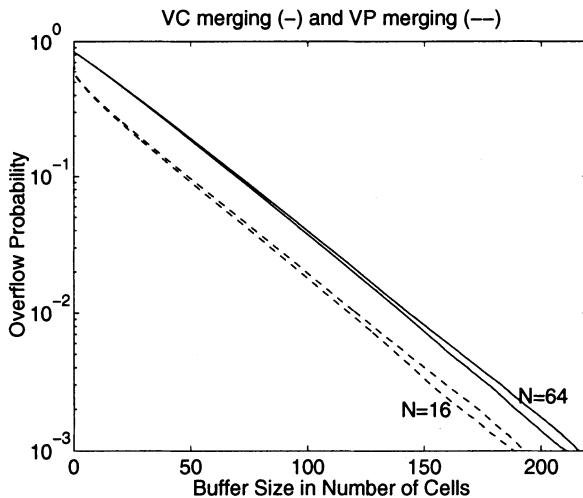
Figure 11 shows the results of the same simulation for  $l = 70\%$ ,  $k =$



**Figure 11** Simulations for  $l = 70\%$ ,  $k = 1.2, 1.5, 16$ , bursty arrival process and  $E = 30$ .

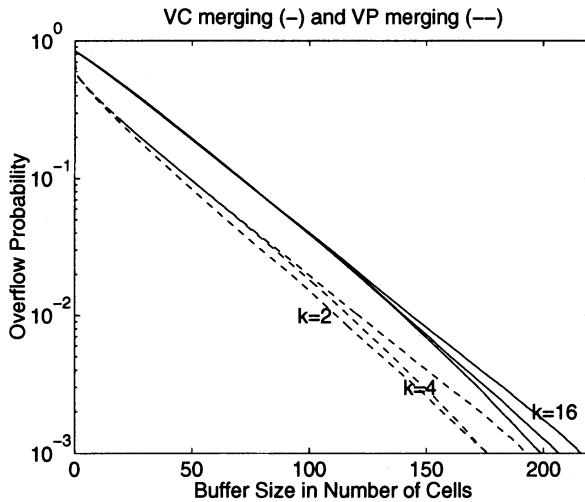
1.2, 1.5, 16 and an increased mean packet size of  $E = 30$ . Compared to Figure 8 we again observe a shift of the critical  $k$  from a value of about  $k = 1.2$  to a slightly larger value. Concerning the additional buffer requirement for VC merging, the same observations were made as in the case of load  $l = 90\%$ . This means that, also at this load, as the packet size increases, the additional buffer requirement increases by the same factor.

Finally we performed simulations for a larger  $N$  ( $N = 64, 128$ ) to assess the influence of a large number of sources associated with one mpt-to-pt connection on the additional buffer requirements of VC merging due to reassembly. An increased number of sources could translate to an increased degree of reassembly. This again would lead to a significantly larger required buffer space for reassembly than for nonreassembly. Figure 12 shows the results for the simulations for  $N = 16, 64$  sources, factor  $k = 16$  and nonbursty traffic. The results obtained also apply in the case of nonbursty traffic. This is explained by Palm-Khintchine's theorem (Heyman and Sobel 1982, p. 156), which states that summing up a large number of iid processes (for instance hyperexponential processes as used for our bursty traffic) results in a process of Poisson type (our nonbursty traffic). As our simulation has 64 sources, each with an iid process for the arrival traffic, we are able to apply this theorem and to simulate nonbursty arrival traffic. All of the corresponding curves in Figure 12 lie close together. Concerning VP merging, as  $N$  increases, the corresponding curves converge because the aggregated arrival process tends to a Poisson one. For VC merging, the buffer requirement does not increase with the number of sources. This is because increasing the number of sources translates to decreasing the arrival rate per source such that the load at the output link



**Figure 12** Simulations for  $N = 16, 64$ ,  $l = 70\%$ ,  $k = 16$ , nonbursty arrival process.

remains constant. This shows that our previous simulation results hold also for a larger scenario with a larger number of senders.



**Figure 13** Simulations for  $N = 64$ ,  $l = 70\%$ ,  $k = 2, 4, 16$ , nonbursty arrival process.

We have investigated the impact of varying  $k$  given large values of  $N$ . Previously we found a critical  $k$  of about 1.1 to 1.3 at  $N = 16$ . Figure 13 shows the results of the simulations with  $N = 64$  and  $k = 2, 4, 16$  with nonbursty traffic. The different curves for VC and VP merging again lie close together and we see no significant difference between the curves belonging to the values  $k = 2$  and  $k = 16$ . Consequently the critical value of  $k$  is smaller than 2. Once again, for values of  $k$  greater than the critical value, the additional buffer requirement for VC merging does not increase.

#### 4 SUMMARY AND CONCLUSIONS

VC merging is likely to become the method of choice to implement mpt-to-pt connections in ATM networks. Because of the cell interleaving problem created by VC merging, reassembly has to be performed in the merging points. The effect of reassembly has been investigated assuming an output queue switch architecture. The results obtained demonstrate that, at high loads and for arbitrary arrival processes, the implementation of VC merging in the switches will not require much additional buffer at the output queues of the switches. In contrast, at low loads, additional buffer is required but this is minimal. Furthermore, it was found that the additional buffer requirement for VC merging is proportional to the average packet size. Consequently, large



packet sizes can result in large reassembly buffer requirements. We further investigated the effect of the speed ratio between switch output port and output link and came to the conclusion that for sufficiently large speed ratio values ( $k > 2$ ) the output buffer requirement for VP and VC merging remain the same, respectively. We found a critical  $k$  which grows with decreasing utilization and also with growing mean packet sizes of the arrival traffic. But it always remains between 1.1 and 1.3 for high utilization of 70% and 90%.

## REFERENCES

- Atkinson, R. (1994) Default IP MTU for use over ATM AAL5. RFC 1626, May 1994.
- Callon, R., Doolan, P., Feldman, N., Fredette, A., Swallow, G. and Viswanathan, A. (1997) A framework for multiprotocol label switching. Internet Draft <draft-ietf-mpls-framework-02.txt> (November 1997).
- Calvignac, J., Basso, C., Droz, P. and Dykeman, D. (1997) Dynamic Identifier Assignment (DIDA) for merged ATM connections. ATM-Forum / 97-0504 (July 1997).
- Chao, H.J. and Smith, D.E. (1992) A shared-memory virtual channel queue for ATM broadband terminal adaptors. *Int'l J. Digital and Analog Commun. Systems*, 5, 29–37.
- Denzel, W.E., Engbersen, A.P.J. and Iliadis, I. (1995) A flexible shared-buffer switch for ATM at Gb/s rates. *Comp. Networks & ISDN Systems*, 27, 611–24.
- Heinanen, J. (1993) Multiprotocol encapsulation over ATM adaption layer 5. RFC 1483, July 1993.
- Heinanen, J. (1997) Multipoint-to-point VCs. ATM-Forum / 97-0261 (April–May 1997).
- Heyman D.P. and Sobel, M.J. (1982) *Stochastic Models in Operations Research*. Vol. 1. McGraw-Hill, New York.
- Iliadis I. and Denzel, W.E. (1993) Analysis of packet switches with input and output queuing. *IEEE Trans. Commun.*, 41, 731–40.
- Laubach, M. (1994) Classical IP and ARP over ATM. RFC 1577, January 1994.
- Mogul J. and Deering, S. (1990) Path MTU discovery. RFC 1191, November 1990.
- National Laboratory for Applied Network Research (NLANR)  
<http://www.nlanr.net/NA/Learn/packetsizes.html>
- Oie, Y., Murata, M., Kubota, K. and Miyahara, H. (1989) Effect of Speedup in Nonblocking Packet Switch, in: *Proc. ICC'89*, Boston, MA, pp. 410–4.
- Piscitello D. and Lawrence, J. (1991) The transmission of IP datagrams over the SMDS service. RFC 1209, March 1991.
- Thompson, K., Miller, G.J. and Wilder, R. (1997) Wide-area internet traf-

fic patterns and characteristics. *IEEE Network*, November/December issue.

Widjaja I. and Elwalid, A.I. (1997) Performance issues in VC-merge capable switches for IP over ATM. *ATM-Forum* / 97-0675 (July 1997).

Widjaja, I., Wright, S. and Chatterjee, A. (1997) Interworking of VP-merge, VC-merge, and non-merge ATM switches in a multipoint-to-point environment. *ATM-Forum* / 97-0748 (September 1997).

## BIOGRAPHIES

**Andreas L. Schmid** received an M.S. degree in Electrical Engineering in April 1998 from the Swiss Federal Institute of Technology (ETH) in Zurich, Switzerland. From October 1996 to March 1997, he was affiliated with the University of Stuttgart, Germany, where he participated in the ERASMUS student exchange program. From November 1997 to March 1998, he was affiliated with the IBM Zurich Research Laboratory, where he did his Master's thesis.

**Ilias Iliadis** received a B.S. degree in Electrical Engineering in 1983 from the National Technical University of Athens, Greece, an M.S. degree in 1984 from Columbia University, New York, as a Fulbright Scholar, and a Ph.D. degree in Electrical Engineering in 1988, also from Columbia University. From 1986 to 1988, he was affiliated with the IBM Thomas J. Watson Research Center in Yorktown Heights, NY, as a work-study student. In 1988, he joined the IBM Zurich Research Laboratory as a member of the switching systems group working on broadband switching. He was responsible for the performance analysis and design of the IBM's PRIZMA switch chip. Currently, he works in the field of ATM-based customer premises networks. His research interests include analysis of distributed systems, performance evaluation of computer communication networks, switching architectures, development of protocols and congestion control schemes, and optimization and network design algorithms. Ilias Iliadis is a member of Sigma Xi, IEEE, and the Technical Chamber of Greece.

**Patrick Droz** studied computer science at the Swiss Federal Institute of Technology (ETH) in Zurich. He received an M.S. in 1992. He then joined the ATM Networking Group at the IBM Zurich Research Laboratory in Rüschlikon, Switzerland, as a PhD student. During this time, he also worked on the design and implementation of the ATM control point for the 8260 campus backbone hub. He received a Ph.D. in 1996 for his dissertation entitled "Traffic Estimation and Resource Allocation in ATM Networks." Since then, he has been working in the ATM Networking group as a research staff member. He is currently working in the area of IP/ATM integration, which involves participation in IETF as well as the ATM Forum.

# A Comparison of ATM Stream Merging Techniques

*M. Baldi, D. Bergamasco, S. Gai, D. Malagrino*

*Dipartimento di Automatica e Informatica — Politecnico di Torino*

*Corso Duca degli Abruzzi, 24 — I-10129 Torino (Italy)*

*Tel. +39-11-5647087 — Fax +39-11-5647099*

## **Abstract**

Multi-layer forwarding approaches (a.k.a. multi-layer switching or routing) which use ATM as transport technology, have proven not to scale enough unless route aggregation is performed. In ATM networks route aggregation implies stream merging: cells from different incoming streams are switched to the same outgoing link and labeled with the same stream identifier. This identifier could be either the whole VPI/VCI pair (VC merging) or only the VPI (VP merging). Stream merging approaches are quite often referred to as VC merging approaches and in this paper we follow this naming convention. The standard way of carrying IP over ATM exploits the ATM Adaptation Layer 5 (AAL5) which does not provide native support for VC merging.

This paper provides an overview of the VC merging problem and presents a review of the most common solutions proposed so far. It also presents CLIMAX, a solution that could fit in different scenarios to solve the VC merging problem.

## **Keywords**

Multi-layer routing, stream merging, VC merging, ATM, Multi Protocol Label Switching.

## **1 INTRODUCTION**

Multi-layer routing techniques are quite often conceived for generic layer 2 and layer 3 protocols; however, their most natural application seems to be the combination of ATM (*Asynchronous Transfer Mode*) and the TCP/IP protocol suite in order to benefit from the performance of the former and the well-known

properties of the latter. Multi-layer routing techniques have proven not to be sufficiently scaleable [SCALE] if the ATM network does not allow Virtual Connections (VCs) to be *merged*, i.e., cells from different incoming VCs to be switched to the same outgoing link using the same Virtual Path Identifier/Virtual Channel Identifier (VPI/VCI) pair. This capability, known as *VC merging*, allows multipoint-to-point VCs to be implemented. VC merging is crucial also to the implementation of scaleable group multicast over ATM, since it requires multipoint-to-point VCs too. This paper focuses on the application of VC merging to multi-layer routing, but most of the drawn considerations apply also to the exploitation of VC merging in ATM multicast.

In fact, VC merging is not the only possible solution to improve scalability in multi-layer routed (or switched) networks. In this paper many solutions that improve network scalability performing VP merging, instead of VC merging, are described. Thus, *Stream Merging* would be the best term to address the whole set of techniques; nevertheless, since the term VC merging has been traditionally used, we follow this naming convention in the paper.

VC merging cannot be performed by common ATM switches when higher layer packets are transmitted using the services provided by the *ATM Adaptation Layer 5* (AAL5) [ITU\_AAL], as recommended by most of the proposals for carrying both data and multimedia traffic over ATM networks. In fact, AAL5 relies on the ATM layer delivering all the cells over a VC in the same order they were sent and without misinserted cells. Instead, when VCs are merged by a switch, cells belonging to different VCs get mixed together and are not distinguishable any more. Many different approaches for supporting VC merging have been proposed so far, but none of them has still proven to be the best in every situation. Each of them is particularly suitable for a specific network environment and for specific needs. In Section 2 the proposals appeared so far are briefly described and then compared. Conclusions are drawn in Section 3.

## 2. REVIEW

After providing a framework for classifying VC merging approaches, their advantages and drawbacks are described highlighting:

- hardware and software changes required in both core ATM switches and devices at the edge of the network;
- performance in terms of delay, jitter, throughput, and buffer requirements;
- specific problems.

A comparison of the different approaches is presented in Section 2.11.

### 2.1 Classification of VC merging approaches

Various VC merging approaches have been proposed in the recent past; some of them present many similarities and they can be broadly classified in three

categories according to the philosophy adopted to solve the problem, as shown in Figure 1:

- approaches based on avoiding cell interleaving;
- approaches based on VP switching;
- approaches based on AAL5 modification.

Approaches based on avoiding cell interleaving cause intermediate switches not to forward cells belonging to different packets simultaneously on the same output VC. All the cells belonging to the same packet are gathered and then forwarded all together. Approaches based on VP switching adopt the VCI to identify the packet to which a cell belongs. Finally, approaches based on AAL5 modification introduce an identifier in the cell payload and use it in order to discriminate among cells carrying different packets and traveling on the same VC. As far as the two last approaches are considered, they could be further subdivided into two categories, according to whether the identifier is associated to a packet or a sender.

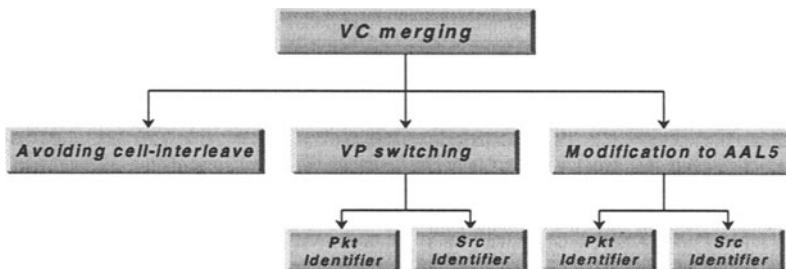


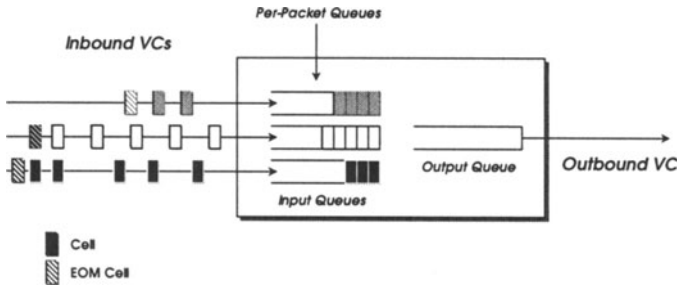
Figure 1 - A classification of VC merging approaches.

## 2.2 MPLS Proposal

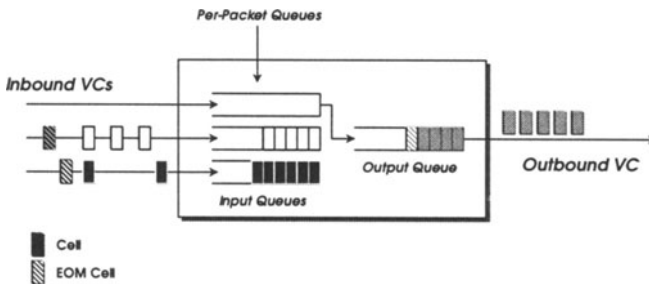
The IETF's *MultiProtocol Label Switching* (MPLS) working group proposes to avoid cell interleaving [MPLSARCH]. ATM switches are modified to implement a special queuing policy for incoming cells traveling on merged VCs. Each switch queues all the cells belonging to a packet until a cell with the *End Of Message* (EOM) bit<sup>1</sup> set (an EOM cell, for short) is received. This indicates that a whole packet has been completely received and buffered. Then, all the cells are transferred to the output queue for transmission. This mechanism avoids that cells belonging to different packets get interleaved on the output link.

---

<sup>1</sup> The EOM bit is set by a transmitting AAL5 entity to identify the last cell of a packet.



(a)



(b)

Figure 2- Cell buffering in the MPLS approach.

Figure 2 schematically shows the behavior of a switch implementing the MPLS approach. In Figure 2(a) no packet is being forwarded on the merged (outbound) VC, because none of the input buffered packets has been completely received. Cells belonging to incoming packets are being queued until their EOM cell is received. When the EOM cell of the gray packet is received, all the cells of the gray packet are transferred to the output queue at once, as shown in Figure 2(b). Note that even if some cells of the white and black packet have reached the switch before the gray ones, they will wait in input queues until their EOM cell is received, i.e., the whole white packet will be transmitted after the gray one.

AAL5 is not modified and ATM switches are not required to parse the cell payload. Even though connection endpoints do not need any change, this approach modifies the forwarding paradigm of switches and this, in turn, implies hardware modifications in ATM switches. Messages are not forwarded cell by cell and thus switches do not feature the latency properties characterizing ATM. However, since packets are not required to be completely reassembled, the MPLS approach demands less processing and introduces shorter latency than packet forwarding at intermediate switches. The extra buffer capacity and the per packet queuing needed in ATM switches could limit scalability.

## 2.3 Simple and Efficient ATM Multicast

*Simple and Efficient ATM Multicast (SEAM)* [SEAM] is very similar to the MPLS solution in buffering incoming cells until the EOM cell is received. Nevertheless, it aims at increasing throughput by forwarding cells immediately (before arrival of the EOM cell) when the output link is idle (*cut-through*). Figure 3 shows how SEAM works; the output queue being empty, the switch immediately forwards the cells of the first packet it began receiving, i.e., the cells of the white packet in Figure 3(a). This prevents cells belonging to other packets from being forwarded; as shown in Figure 3(b), even if the EOM cell of the black packet is received, it waits in the input queue until the EOM cell of the gray packet has been moved to the output queue.

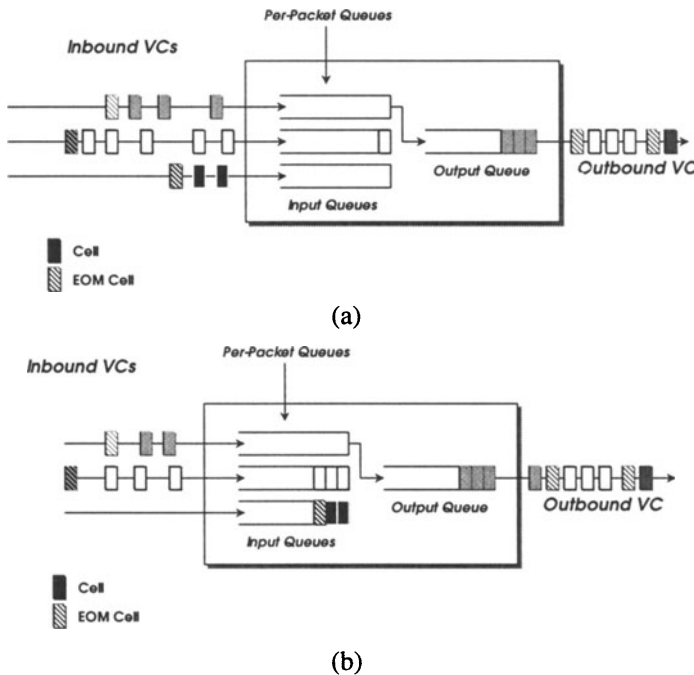


Figure 3 - SEAM approach to VC merging

If the EOM cell of the packet being switched gets lost, cells belonging to other packets are blocked waiting for that EOM cell. A timer is used to overcome this problem; its duration is crucial and impacts significantly buffer requirement into switches. It should be determined on the basis of the bandwidth of the merged VC, the capacity of the links, and the load in the network.

SEAM shares most of the MPLS characteristics and it is not clear if performance is really improved. Short-cutting packets can imply longer latency for short packets (such as TCP acknowledgment messages), when a long packet is being

forwarded. Furthermore, it is hard to determine a suitable timer value since it depends on many parameters; if it is too short, good packets could happen to be discarded during congestion, and if it is too long, it could seriously affect buffer requirement, latency and jitter.

## 2.4 Cell Re-labeling at Merge-points

CRAM (*Cell Re-labeling at Merge-points*) is essentially an optimization of SEAM. It introduces a new type of *Resource Management* (RM) cell, which carries the multiplexing information. When cells are received from more than one input link for the same merged VC, switches suspend forwarding on the merged VC until a reasonable number of cells has been gathered. Then, the cells arrived from different sources are sent on the outgoing link without interleaving them, i.e., grouped according to their source<sup>2</sup>. The trains of cells are preceded by an RM cell which contains a list of *Source IDentifiers* (SIDs) uniquely identifying the sender of each following group of cells. SIDs are ordered as the corresponding groups of cells on the merged VC.

Two mechanisms to guarantee SID's uniqueness are envisioned:

- Globally Unique SID Allocation: SIDs are uniquely assigned by a central server or through a distributed mechanism.
- Dynamic SID Allocation: network nodes dynamically remap the SIDs so that only local uniqueness is required.

CRAM is not compatible with current core and edge devices, even if it does not require any actual modification to the AAL. It requires some minimal changes into switches in order to cope with the new type of RM cell and implement its specific queuing mechanisms. Moreover, some mechanism must be used to cope with the assignment of unique SIDs. Finally, the support to Early Packet Discard (EPD) must be re-implemented because it requires to parse the RM cell payload.

## 2.5 Improved VP Switching and Merging

The improved (or extended) VP switching [VPMERGE] has been proposed by the ATM Forum and it can be categorized among VP switching based approaches. It consists in merging ATM Virtual Paths (VPs). Cells belonging to packets coming from different sources are discriminated through a VCI uniquely assigned by each source. Improved VP switching features all the characteristics of ATM cell switching, thus allowing resource reservation and cell scheduling policies to be kept unchanged, not introducing additional delay in the (VP) merging points.

---

<sup>2</sup> The boundary of the groups does not have to coincide with a packet; a group can either contain cells belonging to one or more packets.



Sources must be provided with a method for identifying a unique VCI value which is chosen at connection setup. At least two different categories of VCI assignments could be identified:

- *Server-based*: a central server in the network is responsible of the assignment of unique VCIs.
- *Signaling-based*: VCIs are negotiated by neighbor nodes.

VP merging presents the disadvantage of using a scarce resource, namely VP space, which limits the maximum number of merged connections on the same link. To overcome this problem, the improved VP switching approach proposes to enlarge the VPI field (18 bits) at the expense of a smaller VCI field (10 bits); the total cell label length is kept unchanged. This is not compatible with the standard operation of ATM switches and every cell must contain an indication of whether the switches should use the long or the short VPI field. The most significant bit of the VPI field is used to provide such an indication, thus halving the available VPI space. Moreover, implementation of improved VP switching requires ATM switches to be modified in order to cope with the new partitioning of the VPI/VCI field.

## 2.6 Dynamic Identifier Assignment

The *Dynamic Identifier Assignment* (DIDA) approach [DIDA] is similar to Improved VP Switching technique and it also comes from the ATM Forum. DIDA does not require packet reassembly at intermediate switches or usage of globally coordinated identifiers. DIDA assigns to each message a locally unique identifier which is inserted in the VCI field. Cells are routed according to their VPI, and the VCI is changed by each switch. The switch identifies any new VCI on incoming cells as the beginning of a new message and assigns a new locally unique VCI to the cell when it is transmitted on the outgoing port.

There are two differences between DIDA and Improved VP Switching:

- In the DIDA approach the VPI space, remains unmodified and is consequently smaller than in improved VP switching.
- VCI semantics and assignment are different. In Improved VP Switching the VCI identifies the source of the cell, while in DIDA it identifies a packet (i.e., packets generated by the same source can have different identifiers).

According to DIDA each identifier is assigned to a message only while it is traveling, thus requiring a small identifier space and no global uniqueness of VCIs. As well as Improved VP Switching, DIDA requires some modification to ATM switches which must modify the VCI in each cell, even though they do not use it for routing the cell. The number of merged connections across each port is limited to 4096, because the VPI field is not extended.

## 2.7 Double Identification Label Swapping

Similarly to DIDA, the *Double Identification Label Swapping* (DILS) approach from IETF [DILS] uses a double level of identification for each packet. The first level identifies the destination and the second the source. DILS envisages three options for the location of the identifiers:

1. The VPI identifies the destination and the VCI the source; the network performs VP switching.
2. One half of the VPI/VCI space is used to identify the source and the other half the destination; switches route cells based on the second half.
3. The VPI/VCI identifies the destination and the source identifier is placed in the cell payload; switches do not require any modification since routing is based on VPI/VCI.

DILS needs an auxiliary protocol to assign source identifiers; hardware changes are needed only with options 2 and 3, listed above. Software changes will be needed, when implementing DILS according to option 1. Options 2 and 3 show higher scalability than option 1, because of the larger labeling space available.

Performances are quite similar to those of Improved VP Switching and DIDA; cell switching is performed with neither extra delay introduced nor extra buffer capacity required.

## 2.8 The Sink Tree Paradigm

The *sink tree paradigm* [SINKTREE] is an innovative approach for ATM Local Area Networks (LANs) which is strongly based on VC Merging. Every switch in the LAN is the root of a multipoint-to-point VC (a *sink tree*) connecting it to all the other switches. A set of sink trees provides full connectivity among switches. Special cells called *connectionless cells* are transmitted over sink trees; they are differentiated according to a bit in the VPI field and are handled differently than regular ATM cells traveling over ordinary VCs. When a source host transmits connectionless cells carrying a packet to a destination host, the source switch places these cells on the sink tree associated with the switch of the destination host. The VPI/VCI fields of connectionless cells carry (1) the source and destination switch identifiers in order to identify the sink tree over which cells must travel, (2) the destination host identifier in order to allow the destination switch to deliver the cells to the proper host, and (3) the source host identifier. The latter enables the destination host to distinguish the cells coming from different sources and properly reassemble them even if many sources simultaneously transmit cells to the same destination host and they get interleaved while traversing the sink tree.

Storing all this information in the VPI/VCI fields limits the scalability of the approach. In fact, the length of the source and destination switch identifiers is 8 bits, and the length of source and destination host identifiers is 5 bits. This means that the largest LAN can span up to 256 switches, each having up to 32 hosts

directly connected, i.e., the maximum number of hosts allowed in a LAN is 8192. These numbers sound quite reasonable in a LAN environment, but prevent the scheme to be exploited in a wide area network.

The Sink Tree Paradigm requires switches to be modified to route cells based on the portion of the VPI/VCI field which identifies the destination switch (i.e., the sink tree on which the cell must travel). Moreover, a protocol for building sink trees and accordingly configuring the forwarding tables of switches is necessary. Edge devices need modifications too since the basic principles for VC creation and management have changed.

Even if the Sink Tree approach keeps the cell based forwarding paradigm typical of ATM, it is not suitable to the provision of service guarantees to applications in terms of controlled delay and jitter. In fact, switches cannot discriminate and properly handle the traffic of a specific application in order to provide it with the required quality of service. The finest possible granularity of traffic segregation into switches is the source-destination pair.

## 2.9 AAL5<sup>+</sup>

In [AAL5+] the VC merging problem is solved through a new AAL slightly differing from AAL5. AAL5<sup>+</sup> overcomes the problems due to cell interleaving by marking all the cells belonging to the same packet with a *Message Identifier* (MID). Its value is assigned by sources on a per packet basis and it is randomly chosen in the range [0, 65535] with a uniform probability distribution. Destinations distinguish cells belonging to different packets thanks to the MID field and properly reassemble incoming packets, even if their cells got interleaved. Since the MID is chosen randomly, two or more messages may have the same MID at the same time. If their cells get interleaved the messages are lost because the destination cannot discriminate the cells belonging to the various messages. This phenomenon is called a *MID conflict* or a *MID collision*. MID conflicts are shown to be really rare and thus they are not explicitly handled. The upper layers reveal the incorrectness of packets affected by MID collision and discard them.

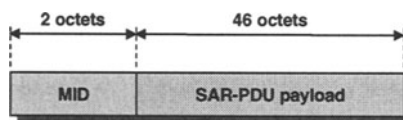


Figure 4 - AAL5<sup>+</sup> SAR-PDU

The MID field, which is 16 bit long, is placed in the ATM cell payload by the *Segmentation And Reassembly* (SAR) sub-layer of the AAL, as shown in Figure 4. Since AAL5<sup>+</sup> uses two octets out of the 48 of the ATM cell payload to carry the MID, its efficiency is lower than AAL5's one (i.e.,  $46/53=86.8\%$  versus  $48/53=90.57\%$ , respectively). AAL5<sup>+</sup>'s efficiency is anyway higher than AAL3/4's one (83%) which inserts a MID in each cell as well and could in principle

represent an alternative to support VC merging. Nevertheless, it is not used to this purpose because the ALL3/4 MID is intended for multiplexing on the same VC different kinds of traffic from the same source (not from different sources). Being AAL3/4 MID shorter (10 bits) than AAL5<sup>+</sup> one, it is not suited to a random assignment, because the probability of MID collisions would be significantly higher.

Even though the efficiency loss introduced by AAL5<sup>+</sup> with respect to AAL5 is not a major issue, the 46 byte payload of the SAR-PDU is not large enough to allow a TCP control message (e.g., an acknowledgment message) to be fully contained into a single cell<sup>3</sup>. This halves the efficiency in the transmission of TCP control messages (e.g., ACK segments) since two ATM cells must be transferred instead of one. Actually, the default encapsulation method of IP packets over ATM networks, requires an LLC/SNAP (*Logical Link Control/SubNetwork Attachment Point*) header (8 bytes) to be put in front of each IP packet in order to allow for multiplexing of different upper layer protocols [RFC1577] on the same VC. In this case TCP control messages do not fit anyway into the cell payload. Moreover, if IPv6 packets [RFC1883] are transmitted using AAL5, TCP control messages do not fit in a single ATM cell since the IPv6 header is 40 bytes by itself.

## 2.10 CLIMAX

The CLIMAX (*Cell-Interleaved Merged ATM conneXions*) approach [CLIMAX], analogously to AAL5<sup>+</sup>, proposes the exploitation of randomly chosen 16 bit Message Identifiers (MIDs) to allow cell interleaving at VC merging points. CLIMAX encompasses two possible implementations which basically differ in the way the MID is carried into cells.

*AAL5<sup>+</sup> Based CLIMAX* inserts the MID in the first two bytes of the cell payload using the same format proposed in [AAL5+].

*VP Switched CLIMAX* inserts the MID in the VCI field of the cell header. This requires a software modification at the transmitting side of end systems in order to randomly choose a VCI value for the cells resulting from the segmentation of the same packet. VP Switched CLIMAX does not require any modification to the hardware of both network nodes and end systems (or edge devices). ATM switches perform VP switching on CLIMAX merged connections and VC switching on other VCs. This solution has a clear scalability limit due to the small dimension of the VPI field. If switches must support both traditional ATM VCs and CLIMAX merged VPs, a bit in the VPI must be used to differentiate between the two kind of VCs and the space of the merged VP identifiers is consequently reduced. VP Switched CLIMAX is completely transparent to the destination, which will distinguish cells belonging to different packets in the same way AAL5 usually

---

<sup>3</sup> The TCP header (20 bytes), the IP header (20 bytes) and the AAL5 CS-PDU trailer (8 bytes) fit exactly in the ATM cell payload.

does. In fact, cells belonging to different packets arrive on different VCs, unless two different sources transmitting on the same VP have chosen the same MID to identify their packets (i.e., a MID collision takes place).

As well as AAL5<sup>+</sup> (see Section 2.9), CLIMAX does not try to avoid MID collision since, in reasonable operating conditions, the MID collision probability is low and the consequent loss is acceptable [CLIMAX-TR], especially if EPD is implemented into intermediate switches as briefly discussed below.

The SAR sublayer in the receiver (of AAL5<sup>+</sup> or AAL5, depending on the specific CLIMAX implementation) gathers payloads of cells with different MID values in different packet reassembly buffers. When an EOM cell is received, the SAR sublayer delivers the corresponding packet to the upper layer and releases the reassembly buffer associated with it. The memory required by the buffers concurrently used by the receiving entity to reassemble messages can limit the scalability of the approach.

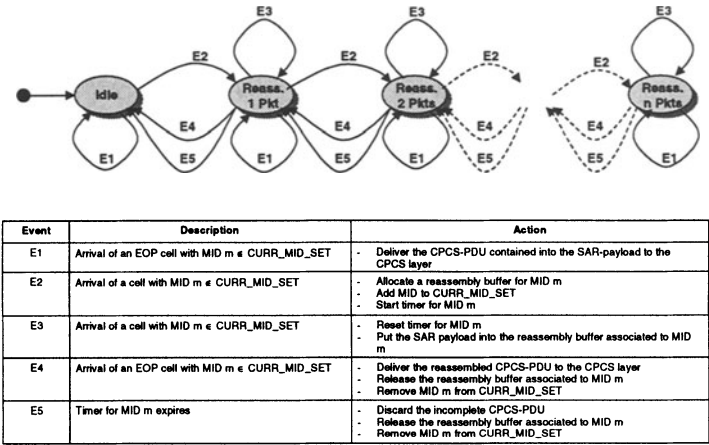


Figure 5 - State-transition diagram of a receiving entity SAR sublayer in CLIMAX

If an EOM cell gets lost, the reassembly buffer associated with one of the packets being reassembled will not be released anymore. If switches implement EPD or a similar packet discarding mechanism, this phenomenon is rare. Since these techniques try to discard only entire packets instead of cells belonging to different packets, they limit the number of incomplete packets delivered to the destination, thus lowering the number of unreleased buffers. Anyway, packet discarding techniques like EPD have not yet reached a large diffusion.

The loss of EOM cells increases the probability of having MID collisions since each unreleased buffer is equivalent to keeping a MID in use until it generates a collision. CLIMAX strictly limits extra buffer allocation exploiting a *buffer release timer*. This reduces MID collision frequency and memory requirements in edge devices. Figure 5 shows the state-transition diagram of a SAR sublayer receiving

entity exploiting the buffer release timer. Each state represents the number of packets currently being reassembled or, equivalently, the number of reassembly buffers simultaneously allocated. A state change occurs when either a cell with a new MID or an EOM cell are received, or the timer associated to a particular MID expires. `CURR_MID_SET` is the set of the MIDs associated with packets being reassembled.

The choice of the duration of the buffer release time-out is critical, as shown by preliminary results of ongoing simulation work [CLIMAX-TR]. A too conservative (too long) timer might result in the need for a large amount of buffer memory and an increased MID collision frequency. This is the reason why the traditional AAL5 time out mechanism which is too loose, is considered not effective. At the opposite end, if the buffer release timer is set too short (i.e., shorter than the maximum cell delay variation experienced in the network), partially reassembled messages may be discarded due to even a single cell which has experienced the maximum delay.

Since buffering requirements at the destinations can affect the scalability of the approach, it is worth comparing CLIMAX with other approaches from this point of view. Destinations implementing CLIMAX allocate a buffer for each message being received on a multipoint-to-point VC. Assuming no loss of EOM cells, the maximum number of allocated buffers equals the number of sources transmitting concurrently on the merged VC. In a real scenario, EOM cells can get lost and reassembly buffer left open, but the exploitation of an effective buffer release timer can keep the number of buffers in use very close to the lossless case. Notice that when multipoint-to-point communications or multi-layer forwarding are performed without exploiting VC merging (i.e., group communications are implemented through a mesh of point-to-multipoint VCs and point-to-point VCs are used with multi-layer forwarding schemes) the total buffering capacity required in each receiver equals the number of sources, i.e., the upper bound for CLIMAX. This is because in each receiving node a different AAL5 entity must be instantiated to terminate each VC, with the consequent allocation of a reassembly buffer. Alternatively, when VC merging is performed by avoiding cell interleaving in merging points (e.g., like in the MPLS approach described in Section 2.2), the buffer space used by CLIMAX receiving entities is needed into switches.

## 2.11 Comparison

In Table 1 a comparison among the three classes of approaches discussed in Section 2 is outlined. The comparison is based on issues relevant to the production and deployment of these schemes (e.g., need for hardware modification).

Hardware changes are needed in either edge or core devices, but most of the approaches do not require both of them. The approaches based on AAL5 modifications require hardware changes in edge devices, while the others usually impact on the core of networks. Notice that usually in wide area networks the ratio

between core and edge devices is 1:20, making it simpler and preferable to change the former. AAL5 compatibility is obviously not granted by approaches based on AAL5 modifications, while it is generally preserved in the others.

Table 1 - Comparison of the VC merging approaches

<i>Category</i>	<i>No cell-interleave</i>	<i>VP switching</i>	<i>Modified AAL5</i>
<b>Examples</b>	MPLS, SEAM, CRAM	Impr.VPswitching, DIDA, DILS (opt.1 & 2), Sink Tree, CLIMAX	DILS (opt.3), AAL5+
<b>Hardware changes in edge devices</b>	No	No	Yes
<b>Hardware changes in the switches</b>	Yes	No (if VPI/VCI partitioning is not changed)	No (if EPD is not needed)
<b>AAL5 compatibility</b>	Yes	Yes	No
<b>Label space for destination</b>	VPI/VCI (28 bits)	VPI (12 bits)	VPI/VCI (28 bits)
<b>EPD Compatibility</b>	Yes (changes needed)	Yes (no changes)	Yes (changes needed)
<b>Buffering required</b>	High	Low	Low
<b>Latency</b>	High	Low	Low
<b>Switching</b>	Pseudo-packet-switching	VP level cell-switching	Pure cell-switching (not for CRAM)
<b>QoS capabilities</b>	Low	Medium (VP based)	High (connection based), lower for CRAM

The label space per destination is an indicator of the scalability of the approach because, if limited, it can reduce the maximum number of edge devices that could be connected to the network.

EPD compatibility indicates if any changes are needed in order to support packet discarding techniques, like EPD. Approaches based on avoiding cell interleave can support EPD, but they will need changes in ATM switches hardware. This is not an added limitation, since hardware changes are needed anyway in this case. Approaches based on usage of a packet identifier - either carried in the VCI field (VP switching) or in cells (AAL5 modifications) - could easily interoperate with current implementations of EPD, but it would be more effective to base packet discarding on the identifier used to support VC merging.

Buffering, latency and switching method are considered significant performance indicators. The first one impacts on cost and complexity of switches while the last two affect the suitability of the approach for controlling delay and jitter. Approaches based on VP switching and those requiring modification of AAL5 present better performances, while approaches based on the avoidance of cell-interleaving could have some limitations, especially when handling traffic other than best-effort.

The Quality of Service (QoS) capability row expresses the suitability of the category of VC merging approach for guaranteeing QoS. Of course, the more cell switching and its properties are preserved, the higher the suitability for providing QoS guarantees.

### 3. CONCLUSIONS

A network using the standard IP over ATM protocol stack in intermediate and end systems does not allow Virtual Connections (VCs) to be merged. This feature is essential to allow for transmission of packets on multipoint-to-point VCs to either solve scalability problems in multi-layer forwarding or group multicast communications. This paper presents a survey of the most common approaches proposed so far to solve the ATM *VC merging* problem. The approaches are grouped into three categories which are compared according to issues relevant to the production and deployment of the required equipment.

Currently, the mainstream approach to solve the VC merging problem in the context of the MultiProtocol Label Switching (MPLS) IETF's working group is based on avoiding cell interleaving in merging points. (Modified) ATM switches buffer all the cells of a packet before starting to forward them; this represents a step away from cell switching towards packet switching.

We consider CLIMAX a very promising approach due to its properties. It is easy to implement and operate, and since it implements traditional cell switching, it is suitable to the provision of Quality of Service (QoS) guarantees. Two CLIMAX implementations are possible: one based on usage of VP switching, the other based on a modification of AAL5 named AAL5<sup>+</sup>. The latter has higher scalability, but



requires hardware changes in edge devices and thus, due to the large number of such devices, it is not an attractive solution for immediate deployment.

We envision a migration towards the massive adoption of VC merging in ATM networks, where the most suitable short term solutions are VP Switched CLIMAX implementation (in small networks) and the MPLS approach (in large networks with a high ratio between the number of edge and core devices and with no QoS requirements). For the long term, the solution which will best combine scalability and cell switching performance is AAL5<sup>+</sup> based CLIMAX.

#### 4. REFERENCES

- [AAL5+] F.Hoymany, D.Mossé, "More Powerful ATM Connectionless Support Using AAL5", Proc. Of IASTED Networks '96, Orlando, Florida, January 1996.
- [CLIMAX] D.Bergamasco, S.Gai, D.Malagrino, "CLIMAX, Cell Interleaved Merged ATM connections", to appear in Proc of 6th Int'l. Conference on Telecommunication Systems, March 1998, Nashville, TN
- [CLIMAX-TR] M.Baldi, D.Bergamasco, D.Malagrino, "ATM VC Merging Techniques: Comparison and Simulative Study", Technical Report, TR-DAI-NET-980320, Politecnico di Torino, Torino, March 1998
- [DIDA] J.Calvignac, P.Droz, C.Basso, D.Dykeman, "Dynamic Identifier Assignment (DIDA) for Merged ATM Connections", ATM Forum Contribution 97-0504, July 1997.
- [DILS] G.Goren, I.Iliadis, P.Droz, "Double Identification Label Swapping (DILS) for Merged ATM Connections", Internet-Draft <draft-droz-dils-arch-00.txt>, July 1997.
- [ITU\_AAL] International Telecommunications Union - Telecommunications Sector, "ATM Adaptation Layer", Recommendation I.363, March 1993.
- [MPLSARCH] E.Rosen et al. - "A proposed Architecture for MPLS", Internet-Draft, <draft-ietf-mpls-arch-00.txt>, August 1997.
- [VPMERGE] R.Venkateswaran, C.S.Raghavendra, X.Chen, V.P.Kumar, "Support for Multiway Communications in ATM Networks", ATM Forum Contribution 97-0316, May 1997.
- [RFC1577] M. Laubach, "Classical IP and ARP over ATM", RFC 1577, January 1994.
- [RFC1883] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 1883, December 1995.
- [SEAM] M.Grossglauser, K.K.Ramakrishnan, "SEAM: Scalable and Efficient ATM Multicast", INFOCOM '97, Kobe (Japan), April 1997.
- [SINKTREE] R.Cohen, B.V.Patel, F.Schaffa, M.Willebeek-LeMair, "The Sink Tree Paradigm: Connectionless Traffic Support on ATM LANs", IEEE Transactions on Networking, V. 4 N. 3, June 1996.

[SCALE] Z.Wang, G.Armitage, "Scalability Issues in Label Switching over ATM", Internet-Draft, <draft-wang-mpls-scaling-atm-00.txt>, July 1997.

## 5 BIOGRAPHY

Born in Cuneo, Italy, on November 9th, 1968, **Mario Baldi** is Assistant Professor of Computer Engineering at Computer and Control Engineering Department of Politecnico di Torino. From April to May 1997 and from July to September 1997, as part of his Ph.D. program, he visited the Computer Science Department at Columbia University in New York, New York, so as to work with Dr. Y. Ofek and Dr. B. Yener on synchronous packet switching techniques for real-time services.

**Davide Bergamasco** was born in Turin (Italy) in 1968. He received his Master Degree in Computer Engineering from Politecnico di Torino, Turin (Italy), in 1995. Currently he is a Ph.D student with the Computer Science Department of the same University. In November 1997 he joined Cisco Systems Inc., San Jose, CA (USA), where he will be working as a visiting researcher until March 1999. His current research interests include, but are not limited to, QoS and performance evaluation of packet switched networks based on computer simulation techniques.

**Silvano Gai** was born in Asti, Italy, on January 20th, 1957. He received the Dr. Eng. degree in electronic engineering from Politecnico di Torino. From 1990 he has been a professor of Computer Network at Politecnico di Torino. His current research topics include: high speed Ethernet, virtual LANs, multilayer switching, IPv4, IP over ATM, IPv6 and Multicast applications. He wrote a book titled "Computer Network: from cabling to internetworking" that is currently used as the classical course book in the Italian Universities. In 1997 he wrote for McGraw-Hill Italia the book "Guide to IPv6". From February 1st, 1997 to October 30th, 1997 he was in sabbatical at Cisco System, San Jose, CA. In 1998 he wrote the book "Internetworking IPv6 with Cisco Routers", published by McGraw-Hill USA.

**Dante Malagrino** (born in 1971) is a Ph.D. student at Politecnico di Torino. He received his Master Degree in 1997, defending a Thesis on "Comparative Analysis of Internetworking Solutions". Since March 1995, he has been working as International Project Manager for the IRISI, a European Project on the Information Society, and he participated to many other European projects. He will carry out a Ph.D. Thesis on "Gigabit Routing" and his main interests include network simulation, high speed network architecture, routing and switching. He published some papers on scientific Italian magazines and on international conferences proceedings, and he participated to the MPLS Working Group of the IETF.

# Integrating Parallel Computing Applications in an ATM Scenario

*Joan Vila-Sallent, Josep Solé-Pareta*

*Universitat Politècnica de Catalunya*

*Jordi Girona 1-3, Mòdul D6 (Campus Nord), 08034 Barcelona  
Catalunya (Spain)*

*E-mail: joanv@ac.upc.es, pareta@ac.upc.es*

## **Abstract**

This paper addresses the problem of supporting communications in parallel computing applications over ATM networks. We propose a mechanism specifically conceived for optimizing the cost-performance tradeoff in fairly long parallel executions. The proposed mechanism relies on a modified version of the loss recovery procedure of SSCOP, which is enhanced by means of a more intensive exploitation of ATM service categories in order to reduce the occurrence of cell loss. For this purpose, we make use of both the UBR and ABR service categories, with ABR being only introduced in the periods of high latency. These periods are determined by periodically monitoring the experienced latency. This approach can achieve equivalent latency as the plain ABR service but with a use of this service of only 30%–70% of the parallel computing traffic, depending on the load of the network and the characteristics of the application.

## **Keywords**

ATM, Corporate Networks, Distributed Parallel Computing

## **1 INTRODUCTION**

The availability of a high-speed network with the flexibility of ATM (Asynchronous Transfer Mode), together with the current evolution of microprocessor technology, is enabling the convergence of communications and computing. In addition, as defined by the ITU (International Telecommunication Union), ATM is the technology that will integrate the whole diversity of network-based services [1]. The combination of these three factors —high-speed networks, microprocessor technology, and integration— is facilitating the development of new applications requiring intensive communications. One of these applications is the support to distributed parallel computing, where a number of

workstations connected to an ATM network can act as nodes of a parallel computing platform. Such environments cannot reach the performance achieved by more expensive, dedicated platforms such as multiprocessors, although they can be a sufficient replacement for many applications [2]. The main bottleneck in network-based parallel computing is experienced in the network itself, and is caused by the delays produced by the protocol processing, the interface with the network, and the processing within the network [3]. These issues occur despite the bandwidth enabled by ATM.

In an integrated environment, communications in parallel computing applications are not limited to LAN environments—which can be adequately supported by Myrinet or Gigabit Ethernet—but are suitable to be extended beyond the local area. The adoption of ATM allows for parallel computing applications to take advantage of an existing network, thus avoiding the underutilization of duplicated resources that would appear with the use of dedicated networks such as Myrinet. Thus, organizations whose parallel computing needs are not very intensive will be able to achieve satisfactory performance with a more efficient exploitation of resources. In this context, parallel computing environments have to subject to a number of conditions in order to achieve sufficient performance with cost-effectiveness. The first condition we assume is that parallel computing applications will share the ATM network with traditional networking applications. Thus, the network architecture will require the presence of mechanisms enabling the support of parallel computing applications that can coexist with equivalent mechanisms for traditional networking applications. In order to preclude the increase in complexity that would arise with the enhancement of ATM with application-specific mechanisms, we consider that the adaption of parallel computing to ATM should be done with mechanisms implemented on top of ATM. Thus, ATM will solely support those services defined in the standards by ITU-T (Telecommunication Standardization Sector of ITU) and the ATM Forum [4].

Many of the applications to be integrated in ATM networks have strong bandwidth and/or delay requirements, as they manage continuous data streams. In these applications, network mechanisms should maximize the network capacity, measured by throughput. In parallel computing applications, however, communications involve the exchange of relatively short pieces of data along a relatively long execution period, so the minimization of communications time has not a tight relationship to network capacity. Thus, communications in parallel computing applications approach to the request-response model, since each task sends data to other tasks and expects other data from them. In this model, per-message overheads set a limit on the achievable performance and therefore, as discussed in [5], latency is a measure that gives a clearer idea about communication performance in parallel computing applications. We consider the latency measure as embedding all per-message communication costs which include, in addition to the costs of overheads and the delays from buffering and scheduling, the eventual need of recovering from cell loss

that results from the need of sharing the network with other networking applications.

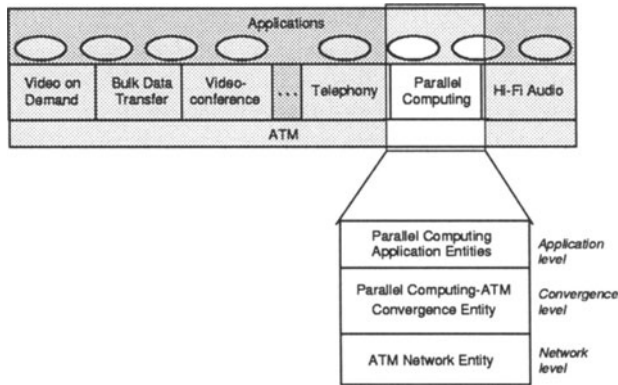
In this paper, we propose a mechanism that enables low latency operation for communications in parallel computing environments over ATM networks integrating different networking applications. In particular, we concentrate on networks spanning outside the local area, where the impact of the applications sharing the network with parallel computing can be more significant. Thus, our mechanism will provide for a strategy to minimize latency degradation caused by the presence of background traffic, which is based on periodically monitoring the latency experienced by communications in real time, in order to achieve cost-effective performance. The rest of the paper is organized as follows: Section 2 presents the general characteristics of the target environments for our mechanism. In Section 3, the particular features of the mechanism proposed in this work are extensively discussed, and their performance is evaluated in Section 4. Finally, Section 5 concludes the paper.

## 2 PARALLEL COMPUTING IN AN INTEGRATED ENVIRONMENT

Most environments to support parallel computing are addressed to operate on multiprocessors or high-speed LANs. Both of these environments can be very expensive when a large number of nodes is required by applications, so they are basically adequate for intensive use of parallel computing facilities. When the need of parallel computing support is not so intensive, it is better to allow for parallel computing environments to extend beyond the local area to provide appropriate scalability to parallel computing applications. In this case, LAN technologies like Gigabit Ethernet are not applicable, so the role of ATM as an integrating technology is more clear. Another important issue outside the local area is the greater influence of network load as more applications are then presumed to share the ATM network. In this paper, we assume that the scenario for distributed parallel computing over ATM will be based on a virtual network comprised by the endpoint hosts supporting the tasks of parallel computing applications, as well as other nodes implementing the procedures providing addressing, connection management, and other signaling functions. In this model, the endpoint hosts support the actual data transfer operations, while the rest of the nodes in the virtual network are in charge of establishing the necessary connections between the endpoint hosts in order to build the topology required for each particular parallel computing application. The signaling procedures operate before and after the actual execution, and are out of the scope of this paper.

Figure 1 displays the architecture of the endpoint part of the ATM-based platform. Data transfer mechanisms for parallel computing are supported in a specific architecture to be integrated with the specific architectures of tradi-

tional networking applications. A proposal for the architecture of the parallel computing service is discussed in [6]. Three levels are considered: (1) *Application level*, which manages the specific requirements of parallel computing applications; (2) *Network level*, containing the functions provided by a particular network technology, as ATM in the present work, and (3) *Convergence level*, which includes those functions that are required for an adequate support of parallel computing applications, and are not provided by the network level as defined in the respective standards.



**Figure 1** Integration of services over ATM.

Communications in parallel computing applications are considered to be based on sequences of elementary data types —integer, float, double, etc.—, called PC-PDUs after “Parallel Computing Protocol Data Units” which are the minimum data structures understood by parallel tasks in a logical sense. Larger structures —arrays, structs, etc.— can be broken into these elementary PC-PDUs. The needs of bandwidth are not very high on average, because communications among parallel tasks are not occurring continuously but an arbitrary period of time can separate the issue of two consecutive messages. Nevertheless, in the particular instants when a message is submitted, very low latency is required in the network in order to minimize the impact of communications on performance. As the mechanisms in the convergence level have to satisfy all these requirements with a full guarantee of data delivery, this paper adopts the specific ATM Adaptation Layer (AAL) proposed in [6], which is based on a modified version of SSCOP (Service Specific Connection Oriented Protocol). SSCOP is a protocol defined by ITU-T in the Q.2110 recommendation [7] for supporting a number of services requiring reliability on top of ATM. This specific AAL replaces AAL5 and improves performance by avoiding the retransmission of more cells than those effectively lost. With this AAL, applications are less sensitive to the network load induced by the rest of

applications sharing the ATM network, and communications achieve better latency performance. This AAL, however, does not rely on any particular ATM service category from those specified by the ATM Forum [4]. In order to optimize performance, we can propose a modified version of the AAL that takes advantage of the features included with these service categories.

The fact that parallel computing applications exchange relatively short messages along a relatively long execution periods makes the use of guaranteed communications services —such as CBR (Constant Bit Rate)— not convenient. Instead, best-effort services as UBR (Unspecified Bit Rate) and ABR (Available Bit Rate) are the most appropriate service categories to support communications in ATM-based parallel computing environments, since their cost will rely mostly on the effective consumption of bandwidth, as opposed to other service categories where the length of connection period will be a more important issue. UBR is the least expensive service category, but the latency can be excessively high due to the cell loss occurring as the network load increases, while ABR is more expensive but faster, as the built-in flow control mechanism allows to achieve lower latency thanks to the fewer retransmissions needed.

In addition, because of the long execution periods, a number of high activity and low activity periods may alternate in the network, as a result of the applications sharing the network. In the periods with low network traffic, the performance of UBR may be sufficient and, as a result, the higher cost of the ABR service category would not be amortized. Thus, for achieving cost-effective performance the data transfer should be conveyed through UBR when the latency experienced in the network and, when latency through UBR is excessively high, data transfer should be moved to an ABR-based connection. A procedure to monitor latency is therefore needed in order to determine when to activate the ABR service category.

### 3 ENHANCED PARALLEL COMPUTING AAL

We focus on the data transfers occurring during execution time by assuming that the necessary connections have been established prior to the execution. As mentioned above, our proposal is conceived to provide cost-effective performance by adapting to the latency experienced by parallel computing communications. In the following we detail the operation of our mechanism, starting with an overview and continuing with a detailed description.

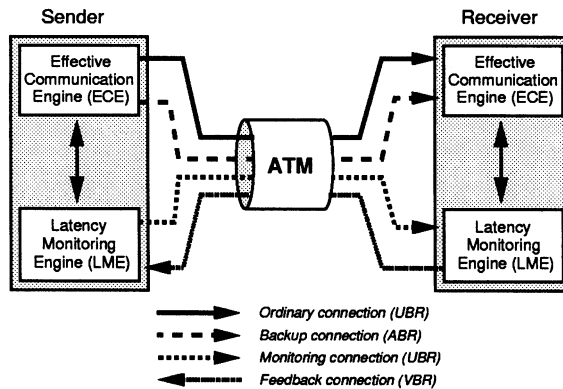
#### 3.1 Architecture

Figure 2 depicts the architecture of our mechanism when the extensions to the Parallel Computing AAL are applied. In each communicating peer, the functionality is contained in two concurrent processes: (1) The *Latency Mon-*

itoring Engine (*LME*), which monitors the latency in the network in order to determine the periods in which significantly high latency is experienced, and (2) the *Effective Communication Engine (ECE)*, which performs the actual data transfers according to the information supplied by the LME. The communications between the engines are served by four connections between each pair of communicating endpoints:

- A UBR-based connection with an unlimited peak rate, used by the ECE to transfer data when latency is low. We refer to this connection as the *ordinary connection*.
- An ABR-based connection with a limited peak rate and a minimum bit rate set to zero, used by the ECE to transfer data when the LME indicates that latency is high. This connection is referred to as the *backup connection*.
- A UBR-based connection like the ordinary connection, which is used by the LME to monitor latency. In practice the same UBR connection is used for both purposes.
- A VBR (Variable Bit Rate) connection with a guaranteed low peak in order to support a fast and reliable delivery of feedback information in the LME.

The adoption of a VBR service category —whose cost is significantly higher than ABR— could compromise the objective for cost-effective performance of our mechanism. However, later in the paper we will observe that the adoption of a VBR-based connection does not significantly impact on performance of parallel computing applications.

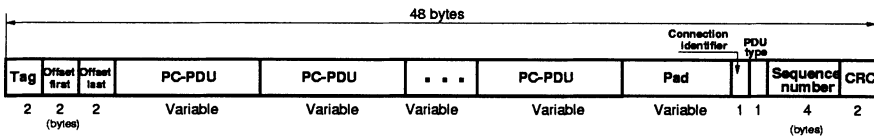


**Figure 2** Mechanisms for extending the Parallel Computing AAL.



### 3.2 The Effective Communication Engine (ECE)

The Effective Communication Engine (ECE) consists of an extension of the Parallel Computing AAL as described in [6] that allows to exploit the information supplied by the LME in order to achieve low latency communications. The mechanism discussed in [6] is based on a modification of the selective retransmission procedure of SSCOP. The modification to SSCOP is addressed to limit the length of the frames to one cell. Thus, unlike standard SSCOP, the amount of retransmitted cells corresponds exactly to the lost cells and, as a consequence, applications become less sensitive to network load. This modification is possible thanks to the short length of PC-PDUs —corresponding to elementary data types such as integer, float, etc., as noted above. In particular, each cell encapsulates as many complete PC-PDUs as possible, so that the data can be integrated with computation as soon as received. In order to avoid the unnecessary overheads involved with the payload length and the 32-bit checksum of AAL5, the mechanism directly replaces AAL5, so it actually operates as a specific AAL. Figure 3 shows the structure of a cell generated by this specific AAL.



**Figure 3** Encapsulation scheme of the specific AAL for parallel computing.

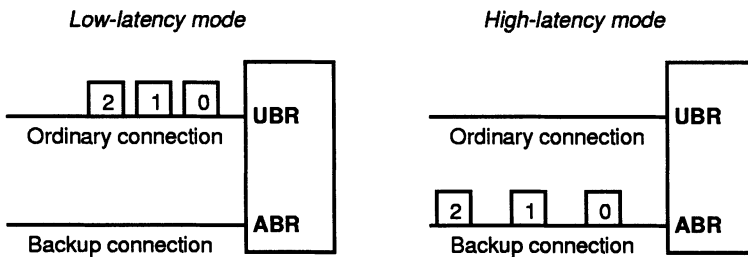
The cell structure shown in Figure 3 includes a significant amount of overhead. Although this overhead obviously leads to throughput degradations, we are more interested in optimizing message latency because of the small significance of throughput on the performance of communications in parallel computing. The overhead includes the following fields:

- *AAL-related fields*, which only includes the CRC field. This 16-bit CRC allows to avoid the unnecessary 32-bit CRC provided in AAL5, which is not adequate for 1-cell AAL PDU.
- *SSCOP-related fields*, which include the Sequence Number and PDU Type fields. These fields are directly inherited from standard SSCOP, but the Sequence Number allows for a larger number space because restricting PDUs to one cell will presumably lead to a higher amount of PDUs.
- *Message-passing library fields*, represented by the Tag, Offset First, and Offset Last fields. They are set to enable compatibility with the PVM (Parallel Virtual Machine) message-passing library [8], which is used by the

parallel programs we have tested. Other message-passing libraries would possibly require different fields.

- **Connection management fields**, which include a Connection Identifier than supports an additional addressing level together with the VCI/VPI fields, in order to facilitate the implementation of a virtual network supporting parallel computing communications.

The ECE enhances the Parallel Computing AAL described in [6] by considering two operation modes: *low-latency mode*, and *high-latency mode*. The extension to the AAL applies essentially to the high-latency mode, which is activated when the LME detects a significant growth in the latency experienced in the ordinary connection. In this case, the backup connection is enabled, so that the cells transmitted on the original connection are switched to the backup connection in order to minimize the impact of cell loss on performance. Figure 4 outlines the operation of the ECE.



**Figure 4** Operation of ECE's latency modes.

*(a) Low-latency mode*

In this mode, the operation of the ECE reduces to the mechanism of the Parallel Computing AAL just outlined. The transfers of data take place over the ordinary connection, so a UBR service is used. Latency monitoring by the LME takes place also over this ordinary connection using a UBR service.

When the receiver part of the LME detects that a monitoring cell has been lost, or when the ECE itself considers that the measured latency is high —i.e. it exceeds a threshold  $T_M$ , the ECE activates the high-latency mode. For this purpose, it issues a new control frame, called LSTAT, which is equivalent to a STAT frame but contains also a time stamp corresponding to the instant when the offending monitoring cell was issued from the sender. LSTAT frames are sent through the same VBR service as USTAT and STAT frames.

*(b) High-latency mode*

When the sender (in low-latency mode) receives an LSTAT frame, it switches to the high-latency mode and triggers the retransmission of pending data, just as a STAT frame. Then, all cells are issued through the backup connection

only. As in low-latency mode, USTAT frames are generated when detecting cell loss. When the sender receives STAT and USTAT frames, the retransmission will be conveyed by the backup connection only. Thus, the operation of the ECE in high-latency mode is similar to the operation in low-latency mode, except for the fact that the backup connection (over an ABR service) is used instead of the ordinary connection (over a UBR service).

When the latency monitored by the LME falls below a threshold  $T_m$ , the low-latency mode is again activated by issuing an LSTAT frame to the sender, including again the information about the status of received cells as contained in STAT frames. The sender then retransmits the cells through the ordinary connection only. The threshold  $T_m$  should be lower than the threshold  $T_M$  in order to avoid a continuous switching between both modes. In all cases, latency is monitored by the LME over the ordinary connection only (that is, over a UBR service) regardless of the operation mode.

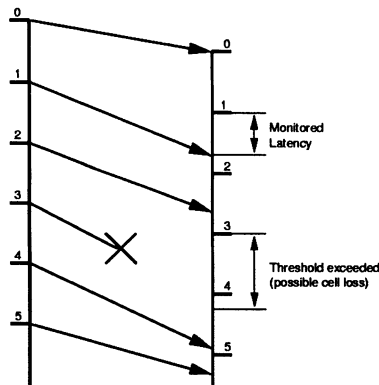
### 3.3 The Latency Monitoring Engine (LME)

The goal of the LME is to provide an estimation of the latency experienced in the ordinary connection. For its implementation we have considered three decisions:

- *Averaging vs. instantaneous monitoring.* Latency can be monitored by computing the average latency over a period of time. This is well suited for applications dealing with large chunks of data, like video and file transfers, but as this procedure has a slow response time, it is not convenient for applications generating more bursty traffic patterns. Therefore, we believe that instantaneous monitoring is a more adequate approach for parallel computing applications.
- *Asynchronous vs. periodic activation.* Latency can be monitored either before a burst of messages or in a periodic fashion. The former case forces the ECE to defer the transmission until the latency is monitored, so it involves a significant amount of latency. In contrast, the latter approach enables the ECE to avoid this delay. For this reason, we believe that a periodic LME is more adequate, despite the extra bandwidth required to support periodic monitoring.
- *The monitoring mechanism.* We can consider the following options: (1) using network-level information; (2) computing the Round Trip Time (RTT); and (3) synchronizing peers and using time-stamped information. The first approach requires the use of a ABR-like network level mechanism providing accessible feedback information, which is not currently standardized within ATM. In the second case, the computed time depends on the latencies of both the monitored connection and the returning path, which are not necessarily equivalent. In the third approach, the experienced la-

tency is monitored by the receiver LME peer, so there is no influence of the returning path on the computed value. As a result, we adopt the third approach as we find that it suits better the requirements of parallel computing communications.

The operation of the adopted approach for the LME is as follows: the sender periodically submits a cell containing a time-stamp. When the receiver gets this cell, it compares its time-stamp to the time the receiver expected to get the cell. The measured latency corresponds to the difference between both times, and then the measurement is passed to the ECE so that it takes the appropriate action, which in the implementation of the ECE discussed above consists of replying to the sender if the monitored latency exceeds a threshold. As the time-stamped cells might be lost, when a certain amount of time  $T_L$  has elapsed since the expected time, the receiver warns the ECE of that circumstance, meaning that a monitoring cell is possibly lost. Figure 5 illustrates the operation with an example. As an enhancement to this basic procedure, the cells issued by the ECE through the ordinary connection are also monitored their latency in order to reduce the response time of the whole mechanism.



**Figure 5** Operation example for the time-stamped LME.

It is important to note that both sender and receiver must be synchronized to each other in order for the measurements to be significant. For this purpose, one of the peers has to report the other one on its current time with a certain periodicity. Thus, we consider two tasks included in the time-stamp LME: (1) *Monitoring task*, which deals with both the periodic and the ECE-originated time-stamped cells; and (2) *Resynchronizing task*, which guarantees that time values are consistent for both communicating peers. We can make use of the different service categories provided by ATM in order to implement these tasks. The Monitoring Task is carried out over the same

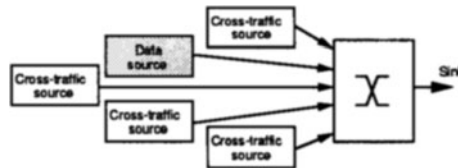
connection as the ordinary data transfers in the ECE, so it is supported by a UBR service. The Resynchronizing Task requires also high priority and, as it is periodic, a CBR service is more adequate. Note that the peak rates for the CBR service should keep low in order to avoid the allocation of an excessive amount of resources. The concrete value of the period depends mostly on the characteristics of the system clocks in both communicating peers, since the more diverging the clocks are, the more frequently the Resynchronizing Task should be activated. In the experiments presented below, we will assume both endpoints as perfectly synchronized and, therefore, no resynchronizing task is considered.

## 4 PERFORMANCE MEASUREMENTS

The goal of the mechanism presented so far is to allow for parallel computing applications to achieve satisfactory performance while keeping the cost not higher than strictly required. To characterize the performance, the average end-to-end latency has been measured in a simple configuration, in order to realize the impact of the mechanism. The cost of the mechanism is also determined and compared to that of the standard ABR service.

### 4.1 Experiment configuration

For moderate network sizes and buffer capacities, the most significant contributions to latency come from the bottleneck links in the ATM network, due to the cell loss and subsequent retransmissions occurring when becoming congested. Thus, the configuration shown in Figure 6 is sufficient for evaluating the performance of the proposed mechanism, and is simple enough to allow for simulations to keep within a reasonable duration.



**Figure 6** Simulated environment.

All the links have a capacity of 155 Mb/s. Two types of sources are considered: one *data source* modeling traffic from a real parallel computing application by means of a trace, and a number of *background sources* modeling traffic from traditional networking applications, by means of ON-OFF sources. The

traffic generated by the data source corresponds to the messages generated by one task of the parallel computing application. In contrast, the traffic from each background source represents the result of multiplexing many sources of traffic from traditional networking applications.

Traces for the data source have been obtained from the execution of parallel codes from the GENESIS benchmark suite [9]. In particular, the considered codes have been *PDE1* and *PDE2*. *PDE1* is a solver of the Poisson Equation on a three-dimensional grid by using red-black successive over-relaxation. *PDE2* solves a two-dimensional Poisson equation using a multigrid method. The traffic generated by *PDE1* consists of relatively long bursts (around 8 KB). In contrast, bursts from *PDE2* are much shorter (50–100 Bytes). As a result, different behavior is expected for each code.

As far as background traffic sources are concerned, the values for the parameters of both the ON and OFF states are exponentially distributed. In the measurements, several sets of values have been used in order to obtain diverse aggregate input rates. In particular, the network utilization  $\rho$  ranges from 0.3 to 1.1, with respect to the output link capacity.  $\rho$  stands for the average network load along the execution period. A value for  $\rho$  greater than 1 indicates that the aggregate incoming traffic is on average higher than the output link capacity. As each background source models the result of multiplexing several sources we do not want a very aggressive background traffic. Thus, the parameters of the ON-OFF models generate a traffic pattern with a burstiness not higher than required to capture the characteristics of multiplexed cell streams. As demonstrated in several papers, for example [10], their burstiness decreases as long as the number of multiplexed sources grows.

The switch is modeled as output-queued. Two priority levels are considered: one for guaranteed service categories (in particular VBR), and the other for best-effort service categories (ABR and UBR). The buffer space is shared by the logical queues associated with each priority level. The buffering scheme is basically drop-tail, except for the case of a full switch buffer, where the arrival of a non-UBR cell forces the dropping of an UBR cell already queued in the switch. The aggregate incoming traffic is arranged in order for the switch to contemplate it as a mixture of UBR and ABR traffic. The ABR scheduling algorithm adopted in the measurements is based on ERICA (Explicit Rate Indication for Congestion Avoidance), fully described in [11]. Table 1 shows the values for the most relevant parameters in the switch and the sources, which in turn are mostly based on the defaults suggested in [4, 11, 12]. Table 2 displays the values for the parameters used in the performance evaluation study presented in this section.

**Table 1** Values for the relevant parameters of the ABR service.

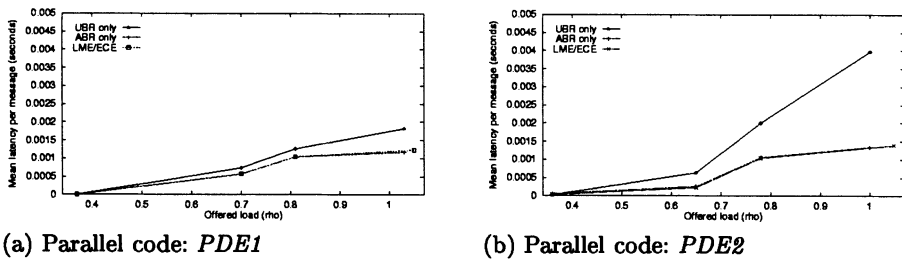
Element	Parameter	Value
Switch	Target Utilization	0.9
	Measurement Interval	30 cells
Source	Nrm	32 cells
	ADTF	0.5 sec
	Peak rate	50 Mb/s

**Table 2** Parameters for our low-latency mechanism.

Parameter	Value
SSCOP POLL interval	0.1 sec
LME monitoring interval	0.1 sec
LME loss threshold $T_L$	0.1 sec
ECE latency threshold $T_M$	0.0001 sec
ECE latency threshold $T_m$	0.00009 sec

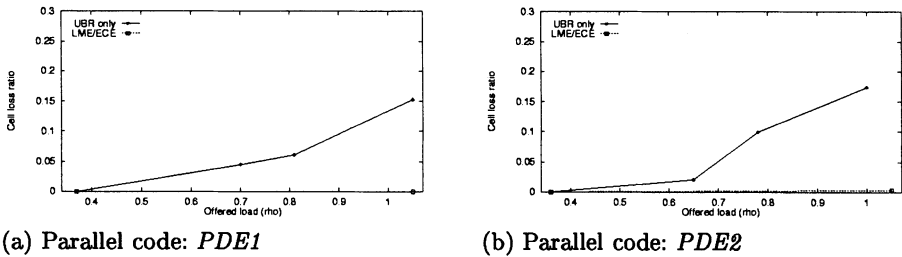
## 4.2 Task-to-task latency

Task-to-task latency is the measure determining the effective impact of communications on the performance of the parallel environment. As we assume that the ATM network is shared with other applications, we expect important variations on performance according to the load of the ATM network. Figure 7 shows task-to-task latency as a function of the different values for the background load. We have compared our proposal for enhancing the Parallel Computing AAL with the AAL without these enhancements, the latter by considering both UBR and ABR as the service categories conveying the data.

**Figure 7** Latency measurements.

According to Figure 7, our mechanism achieves equivalent performance as

that obtained by relying on an ABR service all the time. However, to assess the actual advantages achieved by our mechanism we have to consider other facts, such as the effective utilization of the ABR service and the bandwidth consumption. The relative performance of the measured approaches depends on the particular characteristics of the communications in each application—traffic from *PDE1* is much more bursty than traffic from *PDE2*, as stated earlier. Nevertheless, in the next subsection it is observed that the ABR service is used only by the 30%–70% messages, depending on the application and the network load. Therefore, in addition to equivalent performance, great efficiency in resource exploitation may be achieved.



**Figure 8** Cell loss ratio experienced by the application.

In order to assess the relationship between the performances achieved by both the original UBR-only mechanism and the ECE and the need of retransmissions, we have measured the experienced cell loss ratio with these configurations. The results in Figure 8 confirm that retransmissions are a major cause of latency in ATM-based parallel computing environments, as shown by the close relationship between the ‘UBR only’ curves in Figures 7 and 8, and also that our proposal of ECE succeeds in reducing the amount of required retransmissions, which is characterized in Figure 8 by a cell loss ratio close to zero in the ‘LME/ECE’ case.

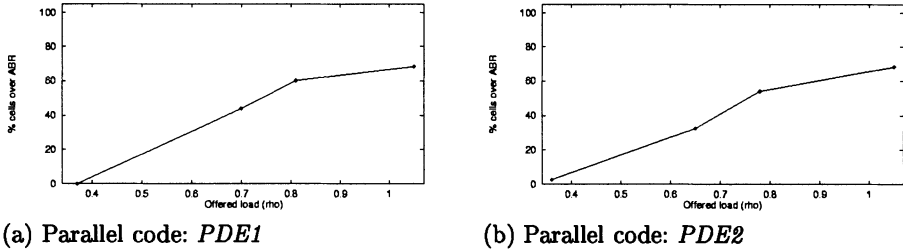
Due to the random component of the background traffic, several repetitions of the latency measurements have been performed. When considering a confidence level of 90%, the maximum radius for the confidence interval is 14% of the mean value in the worst case, which indicates a clear difference between UBR-only results and the rest.

### 4.3 ABR service utilization

We consider the fraction of the cells generated by a parallel task that used the ABR service as a measure of the utilization of this service. As the ABR service requires more resources from the network (a flow control mechanism, some kind of priority, etc.) than the UBR service (which just takes advantage of the bandwidth not consumed by the other service categories, so no particular



resources are allocated for it), the cost of information sent through ABR is also higher.



**Figure 9** ABR service utilization measurements.

Figure 9 displays the results of this measurement. *PDE1* and *PDE2* exhibit different behavior, as expected for the different characteristics of communications. The following observations can be extracted:

- In *PDE1*, the proposed mechanism for the ECE achieves 40% utilization for  $\rho = 0.7$  and 70% for  $\rho = 1$ . These results show a highly cost-effective service achievable by our mechanism. Thus, parallel applications whose communications follow a similar appearance as those of *PDE1* can obtain a performance equivalent to that of the plain ABR service but with a higher efficiency in resource usage.
- In *PDE2*, the ECE achieves a slightly higher utilization of the ABR service—40% for  $\rho = 0.65$ , and 70% for  $\rho = 1$ . In this case, the service remains cost-effective—although slightly less than *PDE1*. The utilization of the ABR service is much less dependent on the application, as opposed to *PDE1*. Thus, applications whose communication pattern is similar to that of *PDE2* can equally achieve cost-effective communications.

In order to realize the effective cost of our mechanism, we should take into account the cost of the VBR service conveying the feedback information. As illustrated below, its performance depends on the network load as well, so we can lose some of the advantage in cost-effectiveness, specially in a highly loaded network.

#### 4.4 Bandwidth consumption

As explained above, our mechanism allows to obtain equivalent performance as that achieved by using exclusively an ABR service, with a fairly low utilization of the ABR service. However, these features are not for free. We have seen that feedback information uses a VBR service, whose cost is higher than the ABR service.

**Table 3** Average bandwidth consumption experienced by *PDE1* (Kb/s).

$\rho$	Service	UBR only	ABR only	LME/ECE
0.7	UBR	278.8	-	154.0
	ABR	-	275.0	122.0
	<i>Total</i>	278.8	275.0	276.0
	VBR	-	-	5.0
1.0	UBR	308.9	-	88.9
	ABR	-	275.0	190.2
	<i>Total</i>	308.9	275.0	279.1
	VBR	-	-	5.1

**Table 4** Average bandwidth consumption experienced by *PDE2* (Kb/s).

$\rho$	Service	UBR only	ABR only	LME/ECE
0.65	UBR	285.9	-	216.0
	ABR	-	289.0	107.6
	<i>Total</i>	285.9.8	289.0	323.6
	VBR	-	-	7.1
1.0	UBR	338.9	-	103.6
	ABR	-	289.0	224.5
	<i>Total</i>	338.9	289.0	328.1
	VBR	-	-	6.0

Tables 3 and 4 reflect the bandwidth consumed in both *PDE1* and *PDE2*, considered as the total amount of bits transmitted along the execution period, by the services carrying the actual data for different two network loads in each case. The results show that, in both cases, the total consumed bandwidth is slightly higher with our mechanism than with the use of ABR only, and the difference is lower in *PDE1*. Using UBR only, as expected, yields the highest consumption due to the amount of retransmitted cells, except for *PDE2* when  $\rho = 0.65$  where the cell loss ratio is not high enough for the rest of mechanisms to become advantageous. Another observation from Tables 3 and 4 is that the fraction of bandwidth spent by the ABR service is closely related to the ABR service utilization displayed in Figure 9.

Regarding the bandwidth spent by the VBR service, we recall that the VBR service conveys the STAT frames, which are periodically generated upon receipt of a POLL frame, as well as USTAT and LSTAT frames which are generated asynchronously. Thus, as expected, the spent bandwidth strongly depends on the cell loss ratio, which in turn is related to  $\rho$ . In particular, the higher the background load, the lower the consumed bandwidth, due to

the increased length of high-latency periods. Note that the significance of the bandwidth consumed by the VBR service is lower than the impact of the ABR service—it is equivalent to 3%–7% of the bandwidth consumption from ABR. Thus, the total cost for the evaluated approach remains advantageous.

## 5 CONCLUSIONS

In this paper, we have described and evaluated a mechanism to integrate communications generated by parallel computing applications in a private virtual network environment based on ATM. This mechanism has been designed to enhance the operation of a novel, specific AAL for parallel computing that was suggested in a previous work, which is based on a modified version of SSCOP. The mechanism presented in this work exploits the service categories provided by ATM. Typically, data applications use an ABR service to reduce the occurrence of cell loss, but the use of a UBR service when the network is unloaded can lead to similar performance. Thus, our mechanism for supporting parallel computing applications uses UBR as the basic transfer service but, when latency experiences a significant increase, an ABR service is introduced. By means of this operation, we achieve low latency in communications and a cost-effective service.

To evaluate the performance of our mechanism, we have undertaken a number of simulation-based experiments. In particular, we have measured the end-to-end latency and cell loss ratio experienced by communications, the utilization of the ABR service category, and the bandwidth consumption, particularly of the VBR service category. In view of the results yielded by these measurements, we observe that (1) the latency achieved by our mechanism is equivalent to the latency experienced when conveying all communication through ABR-based connections; (2) as in the worst case only 70% of cells use the ABR service category, the cost of communications with our mechanism is much lower than the cost inherent to the full use of ABR-based connections; (3) the LME succeeds in determining the high-latency periods, since our mechanism has been able to avoid most of cell loss; and (4) the bandwidth consumption is moderate and the requirements for the VBR service category are sufficiently low, so the cost of communications is not significantly affected. As a summary, our mechanism allows for parallel computing applications that execute for a significantly long period to achieve cost-effective performance.

As introduced earlier, the mechanisms suggested in this work implement only the data transfer part of ATM-based parallel computing environments. Given that we want these platforms to extend beyond the local area, the mechanisms to build and manage parallel computing environments should be defined. In particular, these mechanisms should include a user interface in order to facilitate the platform setup, as well as intelligent load balancing algorithms so that optimal performance can be achieved at each time according to the available resources. For longer term research, we believe that applica-

tions other than parallel computing may also benefit from similar mechanisms and architectures, and therefore these can be adapted in order to advance in the integration of services.

## 6 ACKNOWLEDGEMENTS

This work has been supported by CICYT (Spanish Education Ministry) under contract TIC97-1054-CO3-03.

## REFERENCES

- [1] ITU-T, Recommendation I.121. *Broadband Aspects of ISDN*. Geneva, April 1991.
- [2] T. E. Anderson, D. E. Culler, D. A. Patterson, et al. A Case for NOW (Networks of Workstations). *IEEE Micro*, 15(1):54-64, February 1995.
- [3] K. Castagnera et al. NAS Experiences with a Prototype Cluster of Workstations. In *Proceedings of Supercomputing'94*, pages 410-419, 1994.
- [4] ATM Forum Technical Committee. *Traffic Management Specification, Version 4.0*. Document ATM\_Forum/95-0013R10, February 1996.
- [5] J. L. Hennessy and D. A. Patterson. *Computer Architecture. A Quantitative Approach*. Morgan Kaufmann, 2nd edition, 1996.
- [6] J. Solé-Pareta and J. Vila-Sallent. Network-Based Parallel Computing over ATM Using Improved SSCOP Protocol. *Computer Communications*, 19(11):915-926, September 1996.
- [7] ITU-T, Draft Recommendation Q.2110. *B-ISDN ATM Adaptation Layer - Service Specific Connection Oriented Protocol (SSCOP)*. Geneva, March 1994.
- [8] A. Geist et al. *PVM 3 Users' Guide and Reference Manual*. Oak Ridge National Laboratory, 1994.
- [9] C. A. Addison, V. S. Getov, A. J. G. Hey, R. W. Hockney, and I. C. Wolton. The GENESIS Distributed-Memory Benchmarks. *Advances in Parallel Computing*, 8 (Computer Benchmarks):257-271, 1991.
- [10] J. Solé-Pareta and J. Domingo-Pascual. Burstiness Characterization of ATM Cell Streams. *Computer Networks and ISDN Systems*, 26(11):1351-1363, August 1994.
- [11] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, and R. Viswanathan. *The ER-ICA Switch Algorithm: A Complete Description*. ATM Forum, Contribution ATM\_Forum/96-1172, August 1996.
- [12] R. Jain, S. Kalyanaraman, R. Viswanathan, and R. Goyal. *A Sample Switch Algorithm*. ATM Forum, Contribution ATM\_Forum/95-0178R1, February 1995.

## 7 BIOGRAPHIES

*Joan Vila-Sallent* received his Master's degree and his Ph.D. in Computer Science in 1994 and 1997 respectively, both from the Universitat Politècnica de Catalunya (UPC). After receiving the Ph.D. he joined the Advanced Broadband Communications laboratory (CCABA) of the UPC. Currently he is doing research tasks for this laboratory in R&D projects. Joan Vila-Sallent is member of the IEEE.

*Josep Solé-Pareta* received his Master's degree in Telecommunication Engineering in 1984, and his Ph.D. in Computer Science in 1991, both from the Universitat Politècnica de Catalunya (UPC). In 1984 he joined the Computer Architecture Department of the UPC. Since 1992 he is an Associate Professor with this department. His currently research interests are in ATM Networks, IP over ATM and Optical Packet Networks, with emphasis on traffic engineering, traffic characterization and traffic management. Josep Solé-Pareta is member of the IEEE and the ACM (Sigcomm).

# **Part Five**

---

## **Next Generation Internet**

# Differentiated Services: A New Approach for Quality of Service in the Internet

*Florian Baumgartner, Torsten Braun and Pascal Habegger*

*University of Berne*

*Institute of Computer Science and Applied Mathematics*

*Neubrückstrasse 10*

*CH-3012 Berne*

*Switzerland*

*baumgart@iam.unibe.ch*

*braun@iam.unibe.ch*

*habegger@iam.unibe.ch*

## **Abstract**

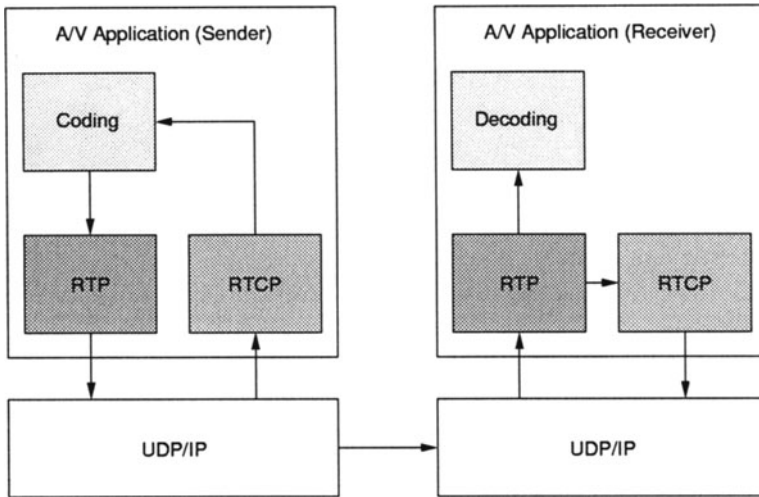
With the grown popularity of the Internet and the increasing use of business and multimedia applications the users' demand for higher and more predictable quality of service has risen. A first improvement to offer better than best-effort services was made by the development of the integrated services architecture and the RSVP protocol. But this approach proved only suitable for smaller IP networks and not for Internet backbone networks. In order to solve this problem the concept of differentiated services has been discussed in the IETF, setting up a working group in 1997. While RSVP classifies packets according to application flow properties, differentiated services are based on the idea that the user negotiates a service profile with his Internet service provider (ISP) for specially marked packets and then transmits marked packets over the ISP network. A further significant difference to RSVP consists in the fact that for scaling reasons the service profile is only negotiated and policed for a set of aggregated flows. This article gives an overview of the activities of the IETF with regards to differentiated services and presents several proposals for the implementation of differentiated services.

## **Keywords**

Differentiated Services, Internet, Quality of Service

## **1 INTRODUCTION**

A central problem of today's Internet exists in the mostly unpredictable service and the often very low quality of transmission. At present there does not exist any satisfactory solution to this problem.



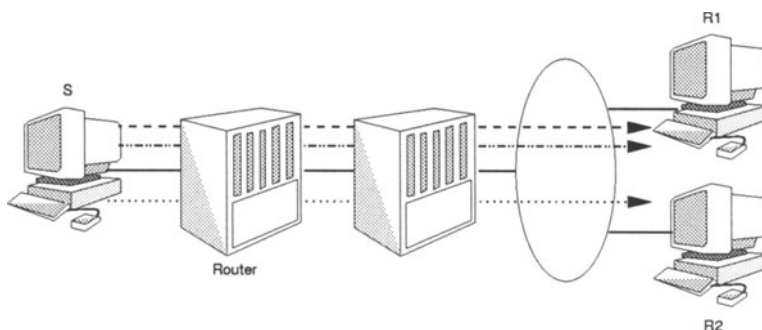
**Figure 1** Monitoring of QoS using RTCP and RTP

A pragmatic approach to achieve good quality of service (QoS) is an adaptive design of the applications to react to changes of the network characteristics (e.g. congestion). Immediately after detecting a congestion situation the transmission rate may be reduced by increasing the compression ratio or by modifying the A/V coding algorithm. For this purpose functions to monitor quality of service are needed. For example, such functions are provided by the Real-Time Transport Protocol (RTP) (Schulzrinne *et al.* 1996) and the Real-Time Control Protocol (RTCP). The receiver in Figure 1 measures the delay and the rate of the packets received. This information is transmitted to the sender via RTCP. With this information the sender can detect if there is congestion in the network and adjust the transmission rate accordingly. This may affect the coding of the audio or video data. If only a low data rate is achieved, a coding algorithm with lower quality has to be chosen. Without adaptation the packet loss would increase, making the transmission completely useless.

## 1.1 Integrated Services and RSVP

Adaptive methods have their limitations when an application requires a certain minimum bandwidth to achieve a reasonable QoS. In these cases a minimal QoS has to be guaranteed by resource reservation. Special applications with real-time requirements depend on resource and bandwidth reservation. This is the reason why the Integrated Services (IntServ) working group defined several services which extend the simple best-effort service: the Controlled Load Service and the Guaranteed Service.

These services are provided for flows i.e. application data streams between end systems. For example three flows exist in Figure 2, two from sender *S* to the receiver *R1* and one flow from *S* to *R2*. Between the sender and *R1* several applications



**Figure 2** Application flows

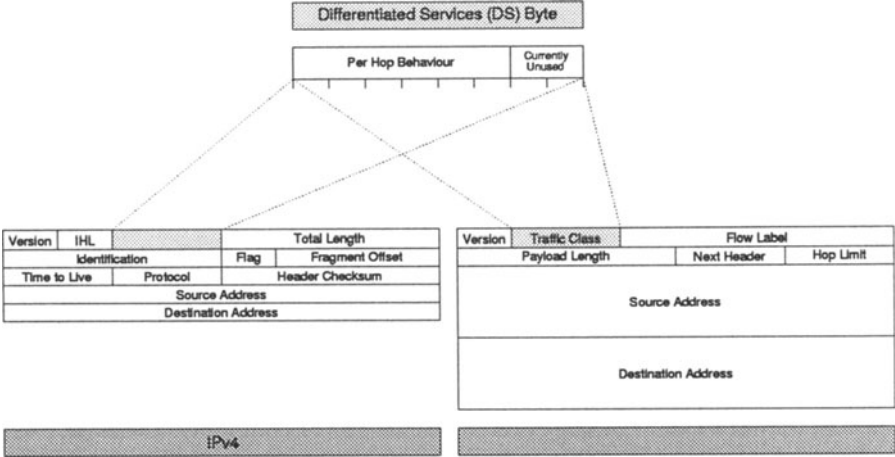
may be active (e.g. a data transmission via FTP and a terminal emulation), or an application may support several flows at the same time (a WWW browser opens TCP-connections to a server).

The resources for a flow are reserved within the end systems and the routers using signaling protocols. For this reason, network elements like routers, nodes and even the operating systems within the end systems have to check whether sufficient CPU time, memory and network bandwidth are available in order to provide a certain service (admission control). Resources then have to be reserved and assigned to the packets of the respective flow (scheduling). Finally, the compliance to the negotiated traffic characteristics has to be monitored (policing).

The Resource Reservation Setup Protocol (RSVP) has been developed as a signaling protocol for resource reservation (Braden *et al.* 1997). RSVP extends the IP protocol stack, i.e. data is transmitted unchanged using IP. It exchanges only signaling information describing the QoS to be given to the TCP/IP or UDP/IP flows. The RSVP resource reservation is receiver-oriented. The receiver generates the reservation message containing the desired service parameters for the received application data flow.

RSVP has been criticized mainly for its limited applicability in large IP networks. The RSVP working group of the IETF has evaluated the applicability of the current version. The flow-based approach is considered as the main problem of RSVP since resources are reserved for every single flow. This cannot be realized with conventional routers if large networks with millions of users and possibly several flows per user have to be supported. Routers are not able to store such a huge number of flow states because of limited memory resources. Secondly, the amount of flows will increase the complexity of packet scheduling in the routers. Scheduling is essential for guaranteed services. A further disadvantage is the lack of standards for accounting and billing, making resource reservation as a result quite unrealistic. For these reasons it is recommended to use RSVP only in small confined networks (Mankin *et al.* 1997).





**Figure 3** DS byte in IPv4 and IPv6

## 2 DIFFERENTIATED SERVICES: BASICS AND TERMINOLOGY

A demand for higher-level services apart from best-effort has been recognized, but these services cannot be realized using the integrated services approach, particularly in large IP networks. The differentiated services model tries to avoid the disadvantages of best-effort networks and the integrated services approach.

The idea of differentiated services is based on the aggregation of flows, i.e. reservations have to be made for a set of related flows (e.g. for all flows between two subnets). Furthermore, these reservations are rather static since no dynamic reservations for a single connection are possible. Therefore, one reservation may exist for several, possibly consecutive connections.

IP packets are marked with different priorities by the user (either in an end system or at a router) or by the service provider. According to the different priority classes the routers reserve corresponding shares of resources, in particular bandwidth. This concept enables a service provider to offer different classes of QoS at different costs to his customers.

The differentiated services approach allows customers to set a fixed rate or a relative share of packets which have to be transmitted by the ISP with high priority. The probability of providing the requested quality of service depends essentially on the dimensions and configuration of the network and its links, i.e. whether individual links or routers can be overloaded by high priority data traffic. Though this concept cannot guarantee any QoS parameters as a rule it is more straightforward to implement than continuous resource reservations and it offers a better QoS than mere best-effort services.

For packet marking the so-called DS byte (for differentiated services) in the header of each IP packet is mapped to the IPv4 Type Of Service octet (TOS) or to the IPv6 Traffic Class octet (Figure 3). Six bits of this byte are used to define the per-hop behavior (PHB) that a packet experiences in each router. The remaining two bits

correspond to the currently unused (CU) field which is reserved for purposes not yet specified and may be assigned later.

The meaning of the individual bits in the PHB field are not yet standardized and are part of ongoing discussions in the Differentiated Services working group (DiffServ) of the IETF. The proposal in (Baker *et al.* 1998) suggests to use one bit for tagging in and out of profile packets and to distinguish service classes with different priorities using the other five bits. Thus, a minimal backward compatibility to the TOS field in IPv4 can be kept. (Nichols *et al.* May 1998) suggests to standardize two different services Default (DE) and Expedited Forwarding (EF) by using two code points in the PHB field. Since the value of the PHB field for a certain service may change at the edge of different ISP networks because of missing standards, it might be necessary to change the value of the PHB field at the border of two networks.

It has to be pointed out that size, meaning and name of the bit fields in the DS byte are subject of further discussions within the DiffServ working group and might change again in the near future. Therefore, the explanations presented here are merely a representation of the status quo of the DiffServ working group. Several sites on the WWW, which are referenced at the end of this text, contain up-to-date information of the exact DS byte definition and should be consulted first of all.

### 3 SERVICES OF THE DIFFERENTIATED SERVICES APPROACH

At present, several proposals exist for the realization of differentiated services. The approach allowing the combination of different services like Premium and Assured Service seems to be very promising. In both approaches absolute bandwidth is allocated for aggregated flows. They are based on packet tagging indicating the service to be provided for a packet.

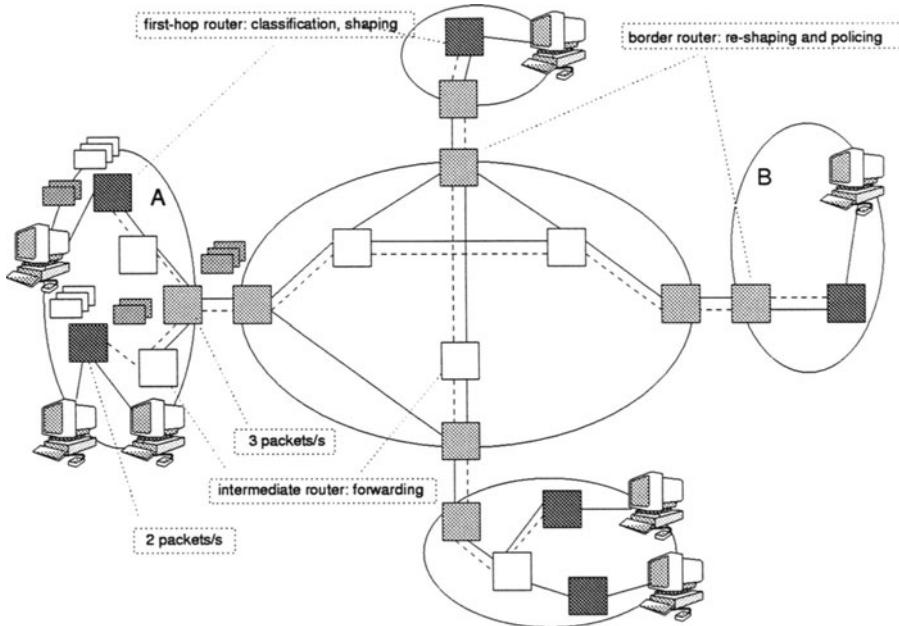
A similar idea is pursued by the Scalable Resource Reservation Protocol (SRP). Flows are aggregated automatically at each link, so that the network does not have to know every single flow. No particular signaling protocol is deployed. Only three different packet types (RESERVED, REQUEST, BEST-EFFORT) are introduced, which differ by the tag in the packet header.

An alternative approach (user-share differentiation, USD) assigns bandwidth proportionally to aggregated flows in the routers (for example all flows from or to an IP address or a set of addresses). A similar service is provided by the Olympic service. Here, three priority levels are distinguished assigning different fractions of bandwidth to the three priority levels gold, silver and bronze, for example 60% for gold, 30% for silver and 10% for bronze.

In the following these services are described in more detail.

#### 3.1 Premium Service

With Premium Service the user negotiates with his ISP a maximum bandwidth for sending packets through the ISP network. Furthermore, the aggregated flow is de-

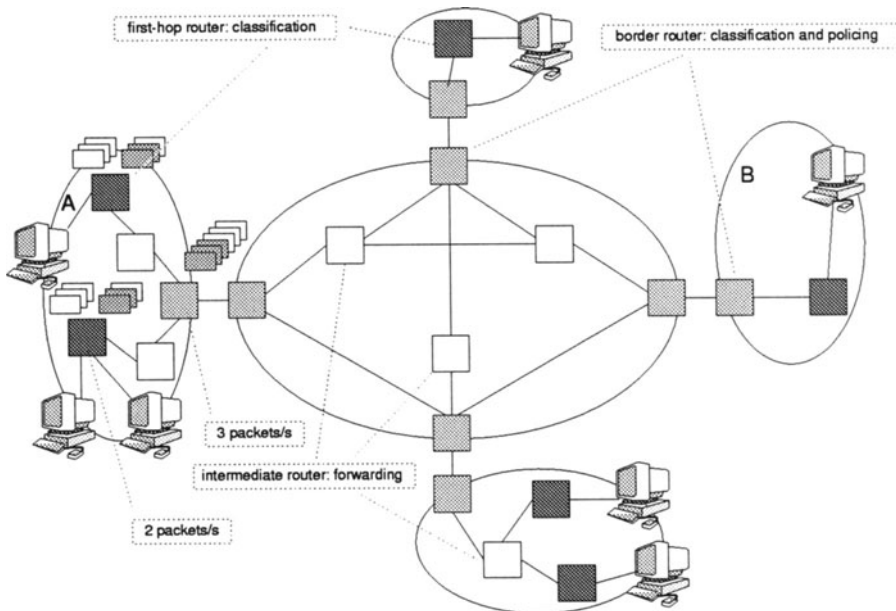


**Figure 4** Premium Service

scribed by the packets' source and destination addresses. In Figure 4 users and ISPs have agreed on a rate of three packets/s for traffic from A to B. The user configures the first-hop router in the individual subnet accordingly. In the example above a packet rate of two packets/s is allowed in every first-hop router as it can be expected that no two end systems will use the full bandwidth of two packets/s at the same time.

First-hop routers have the task to classify the packets received from the end systems, i.e. to analyze if the Premium Service shall be provided to the packets or not. If yes, the packets are tagged as Premium Service (P-bit) and the data stream is shaped according to the maximum bandwidth. The user's border router re-shapes the stream (e.g. three packets per second) and transmits the packets to the ISP's border router, which performs policing functions, i.e. it checks whether the user's border router remains below the negotiated bandwidth of three packets/s. If each of the two first-hop routers allows two packets/s, one packet per second will be dropped by shaping or policing at the border routers. All first-hop and border-routers own two queues, one for packets with the P-bit set and one for all other (see Figure 4). If the P-queue contains packets these are transmitted prior to others. The implementation of two queues in every router of the network (ISP and user network) equals to the realization of a virtual network for Premium Service traffic.

Premium Service offers a service corresponding to a private leased line, with the advantage of making free network capacities available to other tasks, resulting in lower fees for the users.

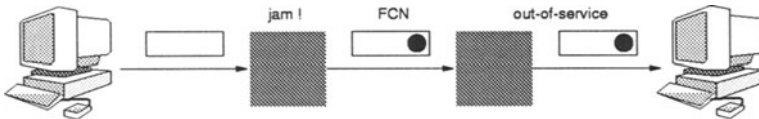


**Figure 5** Assured Service

### 3.2 Assured Service

A potential disadvantage of Premium Service is the weak support for bursts and the fact that a user has to pay even if he is not using the whole bandwidth. The Assured Service tries to offer a service which cannot guarantee bandwidth but provides a high probability that the ISP transfers high-priority-tagged packets reliably. The definition of concrete services has not yet happened, but it is obvious to offer services corresponding to the controlled load service. The probability for packets to be transported reliably depends on the network capacity. An ISP may choose the sum of all bandwidths for Assured Service to remain below the bandwidth of the weakest link. In this case, only a small portion of the available capacity may be allocated in the ISP network. An advantage of the Assured Service is that users do not have to establish a reservation for a relative long time. With ISDN or ATM, users might be unable to use the reserved bandwidth because of the burstiness of their traffic, whereas Assured Service allows the transmission of short time bursts.

With the Assured Service the user negotiates a service profile with his service provider, e.g. the maximum amount or rate of high priority, i.e. Assured Service, packets. The user may then tag his packets as high priority within the end system or the first-hop router, i.e. tag them with an A-bit (see Figure 5). To avoid modifications in the end systems the first-hop router may analyze the packets with respect to their IP addresses and UDP-/TCP-Port and then assign them the according priority, i.e. set the A-bit for conforming Assured Service packets. The maximum rate of high-priority (A-bit) packets must not be exceeded. This is done by (re-)classification in



**Figure 6** Receiver-oriented realization of Assured Service

the first-hop routers and in the user's border routers at the border to the ISP network. Nevertheless, the service provider has to check if the user remains below the maximum rate for high priority packets and apply corrective actions such as policing if necessary.

For example, the border router at the network entrance will tag the non conforming packet as low priority (out of service, out of profile). An alternative would be to charge higher fees for non conforming packets by the ISP. The tagging of low and high priority packets is done by use of the DS byte.

Bursts are supported by making buffer capacity available for storing bursty traffic. Inside the network, especially in backbone networks bursts can be expected to be compensated statistically.

#### **(a) Receiver-oriented scenarios**

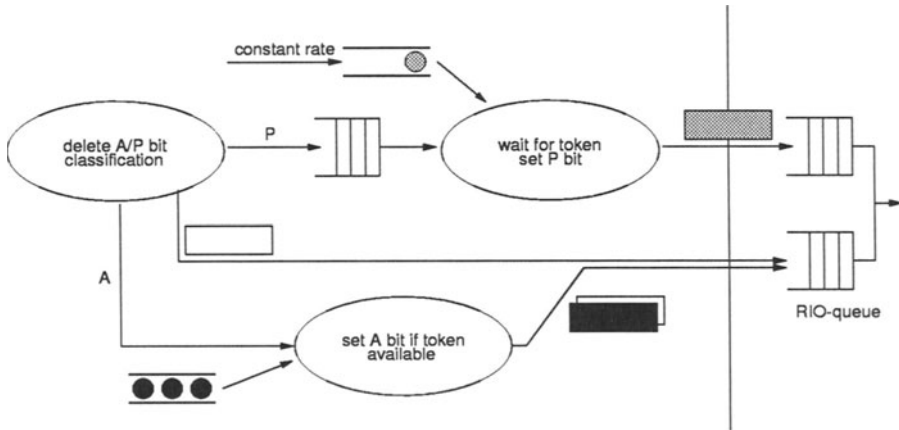
One problem of the Assured Service is the negotiation of the service profile between the sender and the ISP. If an Internet user connects to a WWW server, the receiver should be able to determine the transmission quality and take over the costs. Therefore, the receiver should be able to set up a user profile with the ISP. At the border between the ISP and the receiver's network a border router knows the profile agreement (see Figure 6). This router checks whether the received data flow conforms to the service profile. Otherwise, the ISP's border router sets the forward congestion notification (FCN) bit.

This bit might also be set by routers in the network to indicate a congestion situation. If the packet conforms to the profile the border router resets the bit. For a set FCN-bit the receiver has to slow down the sender's data flow, e.g. by delaying TCP-acknowledgments or by the setting of flow control information. If the receiver does not react, the border router may drop future packets.

#### **(b) Adaptation of applications**

The Assured Service can be combined with the concept of application adaptation. An application can monitor via RTP/RTCP the throughput respectively the loss rate. According to this, more or less packets might be tagged as high priority. If the network is idle the application might transmit best-effort instead of high-priority packets and save costs. On the other hand the application has to increase the number of high priority packets, if a high loss of low priority packets is detected.

The maximum rate of high-priority packets has to be re-negotiated with the service provider, requiring the support of dynamic reconfiguration or signaling.



**Figure 7** First-hop router for Premium and Assured Service

### 3.3 Router implementation for Assured and Premium Service

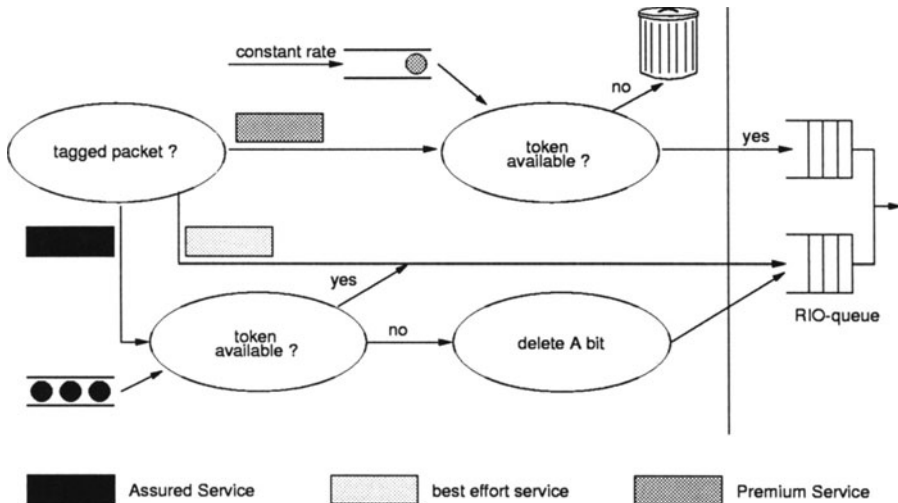
The implementation of Assured and Premium Service requires several modifications of the routers. Mainly classification, shaping, and policing functions have to be performed to the router. These functions are necessary at the border between two networks, for example at the transition of the customer network to the ISP or between the ISPs. Service profiles have to be negotiated between the ISPs similar to the transition to the user.

#### (a) First-hop router

Figure 7 shows the first-hop router function for Premium and Assured Service. Received packets are classified and according to this the A- or P-bit is set if the packet should be supported with Assured or Premium Service. As parameter for the classification source and destination addresses or information of higher protocols (e.g. port numbers) may be used. A pure best-effort packet will directly be forwarded to the so-called RIO-queue. Also, the Assured Service packets get to this queue. The Assured Service packets are checked whether they conform to the service profile. The A-bit will only be kept if the Assured Service bucket contains a token. Otherwise the A-bit will be deleted and the packets are handled as best-effort packets. The RIO-queuing shall guarantee that best-effort packets are dropped prior to Assured Service packets, if the capacity is exceeded.

#### (b) Border router

Similar to the first-hop router an intermediate router has to perform shaping functions in order to guarantee that not more than the allowed packet rate is transmitted to the ISP. This is important since the ISP will check whether the user remains within the negotiated service profile. The border router in Figure 8 will therefore drop non conforming Premium service and reset the A-bit of non conforming Assured Service



**Figure 8** Policing in a border router

packets. Assured Service and best-effort packets share the same queue since both types of packets may belong to the same source. A common queue avoids re-ordering of packets. This is especially important for TCP performance reasons.

### (c) Queuing

An important element in the implementation of Premium and especially Assured Service is a proper procedure for dropping packets in overload conditions. To distribute the available bandwidth fairly among the flows in congestion situations, it is recommended to identify and to drop packets of aggressive data flows.

The fundamental mechanism suggested therefore is the Random Early Detection (RED) mechanism. RED is a new technique for router queue management and is supposed to eliminate disadvantages of traditional queuing mechanisms.

With traditional queuing every supported queue accepts packets as long as possible. If there is no space left in a queue arriving packets are dropped, i.e. the packets at the end of the queue are discarded. This method has two significant disadvantages:

- If bursts arrive at nearly full queues, the likelihood for packets of the burst to get lost is high. But queues are also intended for buffering packets in the case of bursts. Therefore, it is recommended to provide space for those bursts.
- Full queues cause higher delays than queues with lower utilization. Especially for real time or interactive applications higher delays are not desired.

RED (Braden *et al.* 1997) is a mechanism trying to keep the queue length below a certain limit in order to provide space for bursts. This is achieved by dropping packets even if the queue length is relatively small (see Figure 9).

Below the lower threshold no packets are dropped. The more the queue length

exceeds the lower threshold, the higher is the likelihood for dropping a received packet. The dropping is done randomly to prevent dropping the packets of a certain application data flow.

If the queue length reaches the upper threshold, all packets are dropped. With this mechanism the following advantages shall be achieved:

- Bursts can be supported better since there is always a certain queue capacity reserved for incoming bursts.
- By the lower average queue length the delays are reduced, providing better support for real time applications.

RED is especially capable of dividing the available bandwidth fairly among TCP data flows, as packet loss automatically leads to a reduction of an TCP data flows packet rate. The situation with non TCP conforming data as for example real-time applications based on UDP or multicast applications without an adaptation or flow control mechanism is more problematic. They have to be treated special to prevent them from overloading the network.

The queuing algorithm RIO (RED with In and Out) (Clark *et al.* 1997) has been suggested for Assured Service implementation. RIO is an extension of the RED mechanism. A common queue is provided for in-profile and out-of-profile packets, but different dropping procedures (dropper) are applied. The dropper for out-of-profile packets (out-dropper) drops discards packets earlier i.e. at a substantially lower queue length, than the dropper for in-profile packets, i.e. for packets with set A bit. Moreover, the dropping probability of the out-dropper increases more rapidly than the probability of the in-dropper (see Figure 9). This tries to keep the dropping probability of in-profile packets low.

For the implementation of different service types routers have to support several queues, e.g. a queue for Assured or Premium Service. Special bits, e.g. in the TOS field or in the traffic class field of IPv4 respectively IPv6 indicate which service shall be provided to the packet.

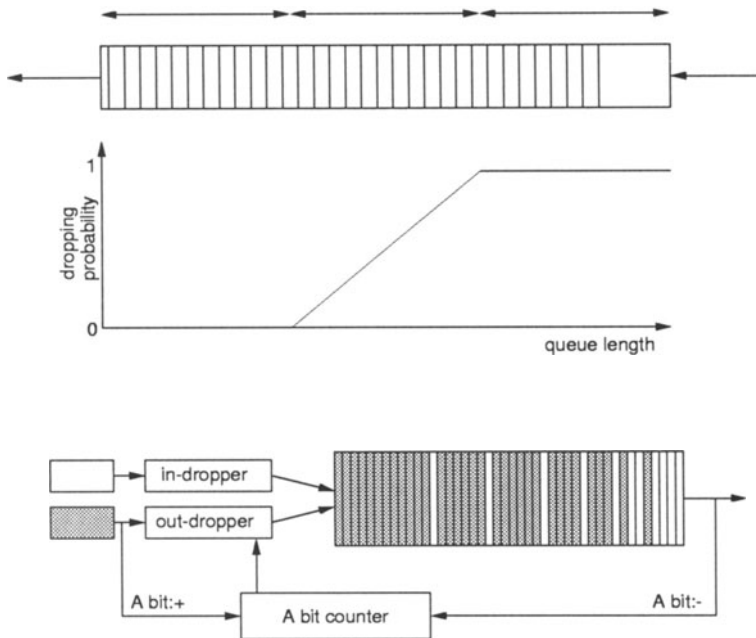
### 3.4 User-Share Differentiation

Based upon packet tagging Premium and Assured Service models can fulfill the stipulated service parameters like bit rates with a high degree of probability only if the ISP network is dimensioned appropriately and non best-effort traffic is transmitted between certain known networks only.

If for instance two users have contracted a bit rate of 1 Mbps for Assured Service packets with an ISP and both wish to receive data simultaneously at a rate of 1 Mbps each from a WWW server which is connected to the network with a 1.5 Mbps link, the requested quality of service cannot be provided.

The User-Share Differentiation approach (Wang 1997) avoids this problem by contracting not absolute bandwidth parameters but relative bandwidth shares. A user





**Figure 9** The queuing algorithm RIO

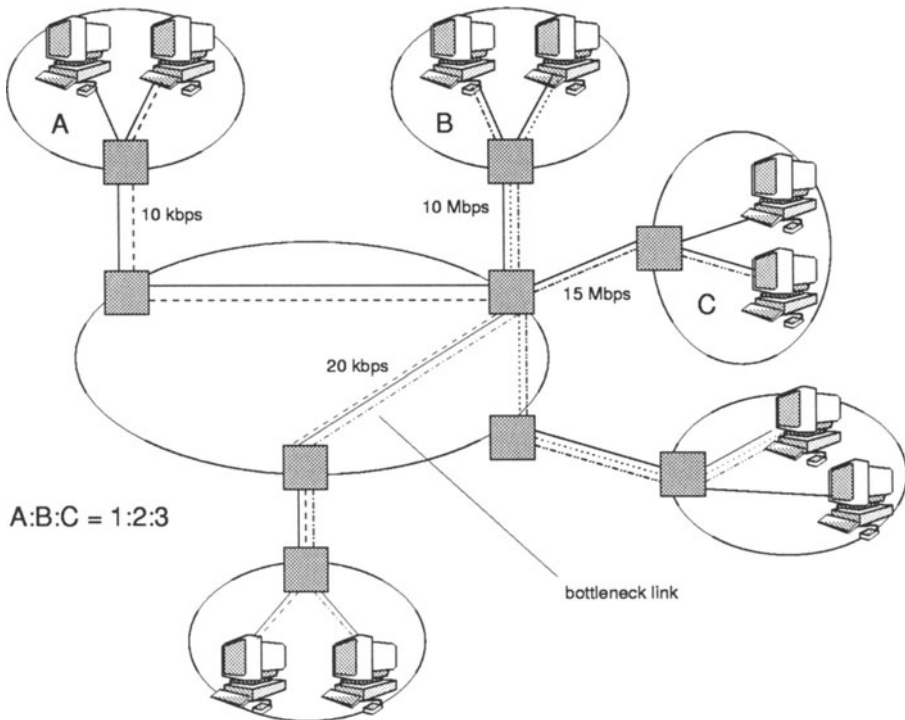
will be guaranteed only a certain relative amount of the available bandwidth in an ISP network. In practice, the size of this share will be in direct relation to the charged costs.

In Figure 10, user A has allocated only half of the bandwidth of user B and one third of the bandwidth of user C. If A and B access the network on low bandwidth links with a capacity of 30 kbps at the same time, e.g. user B will receive a bandwidth of 20 kbps but user C will get merely 10 kbps. If B and C access the same or possibly a different network via a common high bandwidth link with a capacity of 25 Mbps, B will receive 10 Mbps and C only 15 Mbps.

Simpler router configuration is an important advantage of the USD approach. However, absolute bandwidth guarantees cannot be supported.

### 3.5 Olympic Service

The Olympic Service (Nichols *et al.* February 1998) specifies an appropriate service to be deployed within an ISP or a domain. Deployment of this service requires the implementation of a rate-based link share scheduler behavior at each hop. Three service levels are distinguished: gold, silver and bronze. In case of a congested link packets with "Olympic gold" service will get a larger share of the link than packets sent using the "Olympic silver" service which in turn get a larger share than packets



**Figure 10** User Share Differentiation (USD)

with "Olympic bronze" service. When there are no packet flows of gold or silver, packets with "Olympic bronze" service may utilize the entire output link.

By marking packets for a link share flows are classified at a boundary. The exact method of service discrimination is not specified but should be selected in a way that it makes a perceptible difference to customers. A possible configuration of the link sharing could be to allocate 60% for gold, 30% for silver and 10% for bronze, although different configurations could be thought of. Customers do not specify a particular traffic profile for the Olympic Service nor is there admission control, shaping and policing of flows in any way.

### 3.6 Scalable Reservation Protocol

The Scalable Resource Reservation Protocol (SRP) developed at the *Institute for Computer Communications and Applications* (ICA) of ETH Lausanne represents yet another proposal in addition to Assured and Premium Service for a possible implementation of differentiated services in the Internet (Almesberger *et al.* 1997). As indicated by its name much effort has been spent on making the protocol well scalable even for large numbers of packet flows. End systems (i.e. sender and receiver)

play an active part in resource reservation while additional control of the sender's behavior is done at the affected routers. Each router aggregates all incoming data flows and monitors this aggregated data stream in order to estimate the necessary resources (now and in future) at that node.

The so-called estimators play an important role in the process of resource reservation. It is their job to estimate the amount of resources needed for reservation. Estimators are deployed in the sender, the receiver and the routers in between. At the sender it helps to make an (optimistic) prediction on the required reservation of network resources for the data to be transported. The estimator of the receiver computes a (conservative) estimation of the resources actually reserved by the network and periodically sends this information back to the sender.

Without requiring explicit signaling of flow parameters the reservation mechanism consists of a reservation protocol and a feedback protocol which will be discussed in the following.

### (a) The Reservation Protocol

The reservation protocol is deployed from sender to receiver requiring that sender, receiver and all routers in between have implemented this protocol. Three different packet types are distinguished by a tag to be defined in the packet headers.

**REQUEST** Packets marked as REQUEST belong to flows wishing to reserve network resources. If a router forwards such a packet he agrees to accept packets tagged as RESERVED in the future at the same transmission rate. Thus, an implicit reservation at the router takes place.

**RESERVED** If there exists already a reservation at the router and if packets marked as RESERVED arrive at a rate agreed-upon in an earlier stage, the router has to forward them and must not discard them.

**BEST-EFFORT** No reservations exist in the nodes for these packets, and the packets may be deleted by the routers in case of congestion. This service corresponds to today's best-effort service of the Internet.

A sender wishing to make a reservation begins with the transmission of data packets marked by him as so-called REQUEST packets, which already contain the application data. On arriving at a router they are inspected by admission control functions. They monitor the arriving aggregated flow of packets tagged as RESERVED and estimate the amount of local resources needed to maintain a "good" quality of service. These resources consist of the available bandwidth, the buffers' sizes and further local resources of the router. When the router receives a packet tagged as REQUEST for forwarding it has to decide whether the QoS will deteriorate by adding the packet to the existing RESERVED-flow. If this is not the case, the packet, which continues to be marked as REQUEST, can be forwarded, and the estimator of the router has to be accordingly updated.

If the necessary additional resources are not available, the packet is degraded to best-effort service by appropriate tagging before being forwarded. In particular, no

reservations are performed at the router. Packets marked as BEST-EFFORT or REQUEST may be deleted by a router in case of congestion. An end-to-end reservation is only achieved if packets arriving at the receiver are still marked as REQUEST, i.e. resources are allocated at each router on the transport path. By degrading packets marked as RESERVED in case of insufficient resources at a router, a sender cannot get a better QoS by sending only RESERVED-packets.

Reserved resources need not to be released explicitly by the sender. The estimators in the routers will observe an over-allocation of resources after some time after the end of the flow. They will adjust the estimated share of reserved resources in the routers.

### **(b) The Feedback Protocol**

Periodically, the receiver sends back feedback information to the sender containing the arriving rates of REQUEST- and RESERVED-packets measured at the receiver. To this end a special feedback protocol needs to be implemented, e.g. RTP/RTCP (Schulzrinne *et al.* 1996), in order to notify the sender about the current transmission quality. On receiving this feedback information the sender may begin to send packets tagged as RESERVED while observing a transmission rate based on the received feedback from the receiver. If the sender wishes more resources to be allocated for his flow he can keep on sending packets tagged as REQUEST.

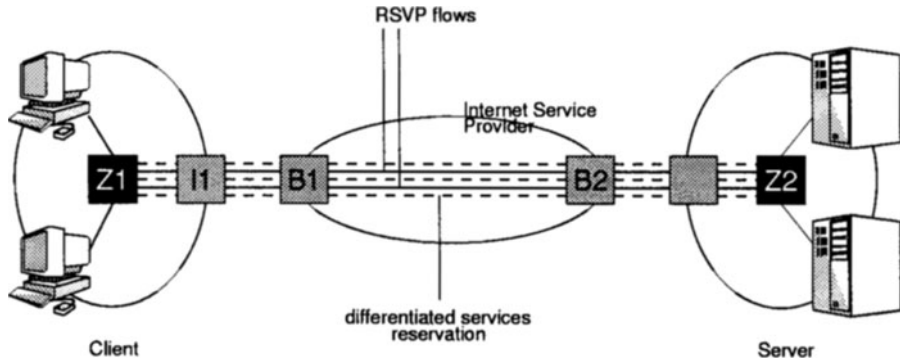
SRP has been tested using simulations, although some topics need further investigation. Policing at network borders and multicasting are not covered in working drafts currently available and are subject of on-going research. The use of SRP for Virtual Private Networks (VPN) is not advised since at each router individual packet flows are aggregated to one large flow which is then treated uniformly. For packet tagging the PHB-field in the DS byte could be used. The necessary code points will be applied for at a future meeting of the DiffServ working group.

## **4 COEXISTENCE OF DIFFERENTIATED AND INTEGRATED SERVICES**

Integrated and differentiated services do not necessarily have to be considered as competing concepts. It is rather advisable to combine both approaches. While differentiated services are recommended for rather large IP networks, the approach chosen for integrated services can be appropriate for limited-size networks, e.g. corporate networks or virtual private networks (VPN).

Both services will be integrated if e.g. VPNs extend over a large IP backbone network. Such a VPN might consist of a client subnet and a server subnet interconnected by a large ISP network, possibly by use of a tunnel. By means of differentiated services techniques both subnets can be linked allocating bandwidth for the traffic between the two subnets.

In such a case it is necessary to map integrated services parameters to differentiated services parameters, similar to the mapping of integrated services parameter



**Figure 11** Reservations with RSVP

to ATM parameters or IEEE 802.1p priorities. In the past such mappings have been defined by the Integrated Services Over Specific Link Layers (issll) working group. In the same way a transformation from integrated services to differentiated services has to be made. In this respect Ford *et al.* (1998) suggest to map guaranteed service to Premium Service and controlled load to Assured Service.

A general framework for the integration of differentiated services and integrated services is proposed in (Bernet *et al.* 1998). The order of events for making an RSVP-reservation in a scenario illustrated in Figure 4 is as follows. First, a sender (server) generates PATH messages. In the server network these messages are processed according to the RSVP protocol by the border router Z2 and other RSVP-routers lying between Z2 and the sender. In the example a reservation for differentiated services has been made between Z2 and Z1, e.g. a Premium Service with a bandwidth of 1 Mbps. In the network between Z2 and Z1 RSVP-messages are transparently forwarded for routers not knowing RSVP. Only the router Z1 processes the PATH-message again. The message arrives at one of the receivers (clients) who can then make a RSVP-reservation using a RESV-message.

This message is processed again by Z1 and Z2. Z2 has to check whether the requested reservation (e.g. 600 kbps) is covered by differentiated services reservation. This is for instance the case if no RSVP-reservation over the differentiated services network has been made yet. If there exists already an RSVP-reservation of 500 kbps between the two subnets, the new reservation of 600 kbps cannot be realized and will therefore be rejected by Z2. Finally, the RESV-message reaches the corresponding sender of the PATH-packet.

When the sender begins to send the real data, Z2 has to do the appropriate mapping on a Differentiated Service. For instance, the DS byte in a packet has to be set to the correct PHB value corresponding to Premium Service if a guaranteed service was requested. Z1 will then reset the DS byte again.

## 5 SUMMARY AND OUTLOOK

At first the differentiated services model seems to be a highly promising concept to provide qualitatively improved services for the Internet since it avoids the obvious drawbacks of the integrated services architecture. In general, however, guaranteed services for application flows following this approach are not possible. It is questionable whether customers will be satisfied with these kinds of services. It seems to be rather interesting to integrate the two concepts of integrated and differentiated services. An important aspect for the success of differentiated services will be if it will be possible to perform appropriate dimensioning of an IP network in a manner that the available bandwidth on all links will be sufficient to forward all differentiated services packets. This presents a very demanding challenge to network planners.

The tasks in the IETF working groups concerning the standardization consist of defining the precise syntax of the DS byte. Moreover, a definition of the management information bases (MIBs) is needed to create a common basis for the configuration of differentiated services parameters in a router. Finally, all the queuing algorithms based on various differentiated services have to be defined in order to implement these services in heterogeneous router environments.

Up-to-date information on the development in the DiffServ working group and related Internet drafts are available on the official homepage of the working group in the WWW. The mailing list dealing with many aspects of differentiated services and the corresponding mail archive offer a close view at on-going discussions and decisions made by the working group in recent time. The URLs to the mentioned resources on the Internet can be found at the end of this text.

Further investigation has to be done on the support of dynamically changing service requirements. Usually, a customer has to negotiate a service contract with the ISP before making use of a service, e.g. by phone, fax, email or WWW form. The agreed-upon parameters then have to be used by the network operators to configure the routers accordingly. Approaches based on *active networking* could be used for this task, e.g. allowing the customer to run configuration scripts on the routers of the ISP. A different approach would be to use a signaling protocol of the requested service parameters, possibly an adapted version of RSVP.

So far, the deployment of differentiated services for multicast services has been hardly investigated. SRP is one of the few approaches where researchers are considering multicasting explicitly. The difficulties essentially lie in the fact that the total need of bandwidth for an IP multicast flow does not only depend on the transmission rate but also on the size of the multicast group and how the individual group members are spread. The latter two criteria however are very difficult to determine in advance. These parameters may dynamically change because of the receiver-oriented IP multicast concept.

For obvious reasons differentiated services could be implemented using networks with QoS capabilities (e.g. ATM). This of course requires a suitable mapping of differentiated services to ATM services. Especially in the area of ATM different concepts of IP switching are going to establish themselves. However, IP switching tries

to bypass IP routers and to forward the packets using switching as often as possible. This in turn may lead to switched packets bypassing shaping and policing functions in the routers, which is inconsistent with the differentiated services architecture. For these scenarios it has to be ensured that either all packets always pass routers with shaping and policing functions or that these functions are realized at so-called ingress and egress routers.

## REFERENCES

- W. Almesberger, T. Ferrari, J.Y. Le Boudec (November 1997) Scalable Resource Reservation for the Internet (work in progress), *Internet Draft*, draft-almesberger-srp-00.txt.
- W. Almesberger, J.Y. Le Boudec, T. Ferrari (March 1998) Encoding of SRP packet types in the DS byte (work in progress), *Internet Draft*, draft-watfjyl-srp-ds-00.txt.
- F. Baker, S. Brim, T. Li, F. Kastenholz, S. Jagannath, J. Renwick (April 1998) IP Precedence in Differentiated Services Using the Assured Service (work in progress), *Internet Draft*, draft-ietf-diffserv-precedence-00.txt.
- Y. Bernet, R. Yavatkar, P. Ford, F. Baker, L. Zhang (April 1998) A Framework for End-to-End QoS Combining RSVP/Intserv and Differentiated Services (work in progress), *Internet Draft*, draft-bernet-intdiff-00.txt.
- B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, L. Zhang (March 1997) Recommendations on Queue Management and Congestion Avoidance in the Internet (work in progress), *Internet Draft*, draft-irtf-e2e-queue-mgt-00.txt.
- R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin (September 1997) Resource ReSerVation Protocol (RSVP) -Version 1 Functional Specification, *Request for Comments*, 2205.
- D. Clark, J. Wroclawski (July 1997) An Approach to Service Allocation in the Internet (work in progress), *Internet Draft*, draft-clark-diff-svc-alloc-00.txt.
- P. Ford, Y. Bernet (March 1998) Integrated Services Over Differentiated Services (work in progress), *Internet Draft*, draft-ford-issll-diff-svc-00.txt.
- A. Mankin, F. Baker, B. Braden, S. Bradner, M. O'Dell, A. Romanow, A. Weinrib, L. Zhang (September 1997) Resource ReSerVation Protocol (RSVP) Version 1 Applicability Statement, *Request for Comments*, 2208.
- K. Nichols, S. Blake (February 1998) Differentiated Services Operational Model and Definitions (work in progress), *Internet Draft*, draft-nichols-dsopdef-00.txt.
- K. Nichols, S. Blake (May 1998) Definition of the Differentiated Services Field (DS Byte) in the IPv4 and IPv6 Headers (work in progress), *Internet Draft*, draft-ietf-diffserv-header-00.txt.
- H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson (January 1996) RTP: A Transport Protocol for Real-Time Applications, *Request for Comments*, 1889.
- Z. Wang (November 1997) User-Share Differentiation (USD) Scalable bandwidth

allocation for differentiated services (work in progress), *Internet Draft*, draft-wang-diff-serv-usd-00.txt.

## 6 INTERESTING WWW-PAGES OF THE IETF

- Official homepage of the DiffServ working group:  
<http://www.ietf.org/html.charters/diffserv-charter.html>
- Mailing list of the DiffServ working group: [diff-serv@baynetworks.com](mailto:diff-serv@baynetworks.com)
- Information and Internet drafts of the DiffServ working group:  
<http://diffserv.lcs.mit.edu>

## 7 BIOGRAPHY

**Florian Baumgartner** M.S. (1997) in information science and physics at the University of Regensburg (Germany). Since 1998 research assistant at the Institute of Computer Science and applied Mathematics with research interests in the simulation and optimization of networks, differentiated services and active networks.

**Torsten Braun** Diploma (1990) and Ph.D. (1993) degree in Computer Science at the University of Karlsruhe. 1990–1994 research assistant at the Institute of Telematics of the University of Karlsruhe. 1994–1995 visiting scientist at the Institut National de Recherche en Informatique et en Automatique (INRIA) in Sophia-Antipolis (France). 1995–1997 guest scientist and senior consultant at the IBM European Networking Center at Heidelberg (Germany). Since 1998 professor of Computer Science at the University of Berne (Switzerland).

**Pascal Habegger** M.S. (1996) in Computer Science at the University of Berne. Since 1996 Ph.D. candidate at the Institute of Computer Science and Applied Mathematics with research interests in network simulations, differentiated services, QoS, ATM.



# Toward a Hierarchical Mobile IPv6

*Claude Castelluccia*

*INRIA Rhone-Alpes*

*ZIRST - 655 avenue de l'Europe*

*38330 Montbonnot Saint-Martin*

*France*

*Claude.Castelluccia@inrialpes.fr*

## Abstract

The IETF Mobile IPv6 protocol provides a mobility management scheme for the Internet. Mobile IPv6 handles macro-mobility and micro-mobility identically. We believe that a hierarchical scheme that separates micro-mobility from macro-mobility is preferable since it would be more scalable. In this paper, we present a mobility management architecture that makes use of the IPv6 Address format hierarchy to provide an efficient and scalable architecture to manage mobility in the Internet. The proposed scheme, which is fully compatible with the IETF solution, differentiates the inter-site mobility management from the intra-site mobility management. The hosts' local mobility is handled with a local, possibly customized, protocol while the global mobility, i.e. across sites, is handled with Mobile IPv6. Our approach has two main advantages. First, the mobility of a host within a site is fully transparent to its correspondent nodes. As a result, the mobility management signaling load is minimized and some of Mobile IPv6 security issues are solved. We show that the signaling load generated by our proposal is at least 69% lower than the Mobile IPv6 one. Second, by differentiating intra-site mobility from inter-site mobility, we provide an architecture that is hierarchical, scalable, flexible and customizable; each site can deploy the intra-site mobility management scheme that is the most appropriate to its particular needs.

## Keywords

Mobile host, TCP/IP networking, Mobile-IP, Mobility Management

## 1. INTRODUCTION

Internet Mobile users require special support to maintain connectivity as they change their point-of-attachment. This support should provide performance transparency to mobile users and should be scalable. Providing performance transparency means that higher level protocols should be unaffected by addition of mobility support. Issues that may affect performance transparency are optimum routing of packets to and from mobile nodes and efficient network transition procedures (Myles, 93). The mobility support should be scalable in the sense that it should keep providing good performance to mobile users and should keep the network load low as the network grows and as the number of mobile node increases. This scalability issue is a very important one in the context of a still growing worldwide network such as the Internet.

The IETF Mobile IPv6 standard, which provides a mobility management scheme for the Internet, does not completely meet these design goals. While it provides performance transparency, we argue that Mobile IPv6 is not scalable. In Mobile IPv6, a mobile node sends a location update to each of its correspondent nodes periodically and any time it changes its point-of-attachment. The resulting signaling and processing load can become very significant as the number of mobile nodes increases. This limitation is the result of the lack of hierarchy in the mobility management procedures of Mobile IPv6. In fact, Mobile IPv6 handles macro-mobility and micro-mobility identically. Since 69% of a user's mobility is local, we believe that a hierarchical scheme that separates micro-mobility from macro-mobility is preferable.

In this paper, we present a mobility management architecture that makes use of the IPv6 Address format hierarchy to provide an efficient and scalable solution. The proposed scheme, which is fully compatible with the IETF solution, differentiates the inter-site mobility management from the intra-site mobility management. The hosts' local mobility is handled with a local, possibly customized, protocol while the global mobility, i.e. across sites, is handled with Mobile IPv6. Our approach has two main advantages over Mobile IPv6. First, the mobility of a host within a site is fully transparent to its correspondent nodes. As a result, the mobility management signaling load is minimized and some of Mobile IPv6 security issues are solved. We show that the signaling load generated by our proposal is at least 69% lower than the Mobile IPv6 one. Second, by differentiating intra-site mobility from inter-site mobility, we provide an architecture that is hierarchical, scalable, flexible and customizable. Our proposal is efficient; it provides optimum routing from and to mobile hosts and improves handoff latency. It is flexible and customizable; each site can deploy the intra-site mobility management scheme that is the most appropriate to its needs.

This paper is structured as follows. In the next section, we introduce some terminology that is used throughout this paper. Section 3 presents the related work including the IETF Mobile IPv6 and its hierarchical derived proposals. Section 4 details our mobility management proposal. Section 5 evaluates and compares the performance of our scheme with the IETF one. Section 7 concludes our paper.

## 2. TERMINOLOGY

The following terms are used in this paper to identify the principal network entities that are of interest to our proposal. A mobile host, *MH* is a node that may move through the Internet. A correspondent host, *CH*, is a host communicating with the MH. The network and the site of a MH when it is not travelling are respectively called the *home network* and the *home site* of the MH. The network and the site that a MH may visit are respectively referred as the *foreign network* and the *foreign site* of the MH. A MH's *Care-of Address* is the global IP address the MH acquires when visiting a foreign network. This address is topologically correct on the foreign network. A MH's *Site Care-of Address* is the first Care-of Address that a MH acquires when visiting a site. When MH is in its home network, it is accessible through its *Home Address*.

## 3. RELATED WORK

In this Section, we present some of the mobility schemes proposed for the Internet. We start with IETF Mobile IPv6 and then describe two of its hierarchical derived approaches, which have been proposed in the context of Mobile IPv4.

The Mobile IPv6 protocol is currently being specified by the *IETF IP Routing for Wireless/Mobile* working group (Perkins, 96b). With Mobile IPv6, each time the mobile node moves from one subnet to another, it gets a new care-of address. It then registers its *Binding* (association between a mobile node's home address and its care-of address) with a router in its home subnet, requesting this router to act as the *home agent* for the mobile node. This router registers this binding in its *Binding Cache*. At this point, the router serves as a proxy for the mobile node until the mobile node's binding entry expires. The router intercepts any packets addressed to the mobile node's home address and tunnels them to the mobile's care-of address using IPv6 encapsulation. The mobile node sends also a *Binding Update* to its correspondent nodes, which can then send packets directly to the mobile node.

While this protocol optimizes the routing of packets to mobile hosts, it is not scalable. As the number of mobile nodes increases in the Internet, the number of Binding messages will also increase proportionally and add a significant extra load to the network.

Caceres and al. have proposed a hierarchical mobility scheme based on Mobile IPv4 that separates three cases : local mobility, mobility within an administrative domain and global mobility in order to reduce the generated signaling load (Caceres,96). This proposal has been made in the context of Mobile IPv4 which uses foreign agents; agents that mobile hosts connect to when they visit a foreign network. (Caceres,96) defines a hierarchy of foreign agents. In this proposal, each subnet that a mobile node could visit has one or more subnet foreign agents, which manage local mobility. On top of those subnet foreign agents, a domain foreign agent manages mobility across the different subnets of an administrative domain. The mobile node's home agent only keeps track of the movement of the mobile node across administrative domain boundaries. As a result, the mobile node's motion within an administrative domain is transparent to the home agent and its correspondent nodes. The hierarchical architecture of this scheme is very interesting but strongly relies on the deployment of foreign agents, which makes it incompatible with the MobileIPv6 protocol.

In the scheme, proposed by Balakrisnan et al. (Balakrisnam,95), packets destined for a mobile node are delivered to the mobile node's home agent using the IETF Mobile IPv4 and are then multicast to multiple base stations in close vicinity of the mobile node. While this approach is hierarchical, we believe that this solution is not very efficient and scalable. In fact, packets destined for a mobile have to transit through the home agent which can be distant from the mobile node's current location. This has the effect of increasing packet delivery latency , handoff latency and the Internet load.

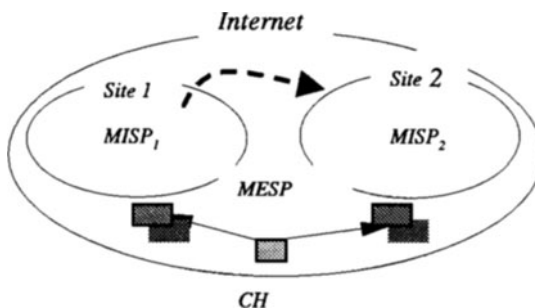
#### 4. A HIERARCHICAL MOBILITY MANAGEMENT ARCHITECTURE

Mobile IPv6 handles local mobility of a host (i.e. within a site or a network) the same way as it handles global mobility (inter-site or inter-network mobility). In fact, in Mobile IP, a mobile user sends location updates to its home agent and its correspondent nodes each time it changes its point-of-attachment regardless of the locality and amplitude of its movement. As a consequence, the same level of signaling load is introduced in the Internet independently of the user's mobility pattern.

We argue that this approach is not scalable and that a hierarchical solution is more appropriate to the Internet. We believe that a user's mobility within a site or a network should be managed locally and transparently to its correspondent nodes. Using such a hierarchical approach has at least two advantages. First, it improves handoff performance, since local handoffs are performed locally. This increases the handoff speed and minimizes the loss of packets that can occur during the transition phase. Second, it significantly reduces the mobility management signaling load on the Internet since the signaling load corresponding to local moves do not cross the whole Internet but is confined to the site or the network. This

hierarchy is furthermore motivated by the significant geographic locality in user mobility patterns. As shown in (Kirk,95), most of a user's mobility is local. According to this study, 69% of a user's mobility is within its home site (within its building and campus) and 70% of all professionals can be classified as mobile. It is therefore important to design a hierarchical mobility management architecture that optimizes local mobility.

We propose a hierarchical architecture that separates local mobility (within a site) from global mobility. Our architecture is hierarchical in two points. First, it separates the local mobility management from the global one. Local handoffs are managed locally and transparently to mobile's correspondent hosts. Second, it clearly separates the protocols managing local mobility from the protocols managing global mobility. In fact, while the hierarchy in the mobility management operations could be performed by the same protocol, we propose to use two different protocols. As illustrated by Figure 1, we define the concepts of *MISP* (Mobility Internal Site Protocol), that manages mobility within a site, and of *MESP* (Mobility External Site Protocol), that manages mobility between sites. The concept of site is quite general. We use the definition of site as it is given in (Hinden,98). A site is a set of networks belonging to the same administrative entity, such as a company or an access provider. Any two hosts of a site must be able to exchange packets without the support of the Internet backbone. A site is connected to the rest of the Internet via one or several interconnection routers. The approach that we propose provides more flexibility to the sites that can deploy the MISP the most appropriate to their needs. A large site can, for example, use a hierarchical mobility management protocol, and add an extra level of hierarchy to the global architecture.



**Figure 1** MISPs and MESP

## 4.1 Inter-site Mobility Management Issues

The Inter-site Mobility Management protocol manages the mobility of hosts between sites. This protocol has to be global to the whole Internet. We propose to use the Mobile IPv6 protocol since it is the current IETF solution and we believe that it is an efficient solution to manage macro-mobility. As with the “regular” Mobile IPv6, a mobile host requires the service of a home agent in its home network. This HA intercepts packets addressed to the MH and forwards them toward the MH’s current Care-of Address.

When a mobile host gets into a new site, it obtains a *Care-of Address*, and communicates it to its home agent and possibly to its correspondent nodes via the emission of a *Binding Update* composed of its *Home Address* and its *Care-of Address*. Thereafter, and as long as the mobile host stays within this site, this Care-of Address, that we call the *Site Care-of Address*, is used in all Binding Updates sent to the Home Agent and Correspondent Hosts. Note that this Site Care-of Address is used in the Binding Updates even if the mobile host moves within the site and gets new Care-of Addresses.

Upon reception of a binding, the HA and CH update their binding list and use the Site Care-of Address specified in the BU to communicate with the MH, in conformance with the Mobile IP protocol specification (Perkins, 96b).

## 4.2 Intra-site Mobility Management Issues

The Intra-site Mobility Management protocol manages the mobility of hosts within a site. As opposed to the MESP, the MISP can differ from sites to sites. They can be customized to each site’s needs. For example, a large site may deploy a hierarchical MISP while a smaller one may use Mobile IPv6. In the next section, we describe a Mobile IPv6-based MISP. This solution results in a 2 level-Mobile IPv6 protocol: one level manages macro-mobility (MESP) and the other one manages micro-mobility (MISP).

### 4.2.1 Mobile IP-based MISP

When a mobile host moves within a site and changes its point-of attachment, it gets a new Care-of address,  $CoA_2$ , and sends a Binding Update, composed of  $CoA_2$  and its Site Care-of Address,  $CoA_S$ , to all of the site interconnection routers<sup>1</sup> of the site and a Binding Update composed of its Home Address and its current Care-of Address,  $CoA_2$ , to its local correspondent nodes. Each interconnection router of the

---

<sup>1</sup> All the interconnection routers of a site could be made accessible via the use of a well-know IPv6 multicast address or via a multicast address communicated to the MHs by the Neighbor Discovery protocol.

site maintains a Binding list with one entry ( $CoA, CoA_S$ ) for each mobile host currently roaming in the site.

When a packet addressed to a mobile host arrives at one of the site's interconnection router, this router searches into its Binding list for an entry whose *Site Care-of Address* field matches the destination address of the incoming packet<sup>2</sup>.

(1) If no entry is found, the packet is routed normally within the site. (2) If an entry is found, the router tunnels the packets to the current (local) Care-of address of the mobile host as specified in Mobile IPv6.

When a host sends packets to a mobile host that is located within its site, it first uses the mobile home address and then switches to the mobile host's Care-of Address as soon as it receives a Binding Update. As a result, if the site is the home site of the mobile host, the first packet is intercepted by the mobile host's home agent and tunneled to its current site address. If the site is not the home site of the mobile host, the first packet is intercepted by one of the interconnection routers of the site and then tunneled to the mobile host's site address.

#### 4.2.2 Others MISPs

Others MISPs could be deployed. For example, the Sony VIP (Teraoka,92) and the Columbia MHP (Bhawat,95), that were designed for small networks, could be good candidates for MISPs. The PIM-based mobility management scheme presented in (Castelluccia,98) could be used for larger sites. A GSE-based approach (O'dell,98) could also be considered. In this approach, the site interconnection routers would dynamically replace the destination address of the packet addressed to a mobile host home address or Care-Of Address with the current Site Address. While this solution prevents from encapsulating packets and makes better use of the local resource, it introduces some security and identification problems.

#### 4.2.3 MISPs Compatibility Issues

An important consideration in our architecture is the MISP compatibility. In fact, it is important, for extensibility reasons, that a mobile host is able to use the different MISPs without having to understand all of them. Therefore, the operations performed by the mobile hosts in the different MISPs have to be standardized.

We propose that the mobile host operations be limited to the emission of Binding Updates to one or several special addresses. These addresses, that can change from MISP to MISP, could be communicated to the mobile host through the IPv6

---

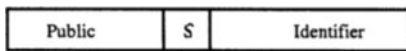
<sup>2</sup> Maintaining these per-mobile host entries is not necessarily a scalability limitation since data structures exist that allow routers to handle long lists of entries efficiently (Sklower,93).

Neighbor Discovery mechanism. For example in the Mobile IP-based MISP this address is the multicast address of the site interconnection routers .

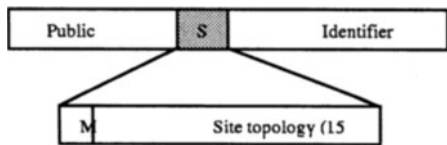
### 4.3 The IPv6 Mobility bit

The site interconnection routers play a central role in our architecture. In fact, they filter all incoming packets to demultiplex packets addressed to mobile hosts from those addressed to fixed hosts. This operation can be expensive if the routers must compare the destination address of all incoming packets against the list of mobile hosts roaming within the site. To minimize this cost, we propose the definition of a *mobility bit* within the IPv6 Addresses to help routers to efficiently demultiplex packets addressed to mobile hosts from those addressed to fixed host.

The *IPng Working Group* has defined an global address format for IPv6, the *Aggregatable Global Address Format* (Hinden,98). This address, that is presented in Figure 2, is structured into a three level hierarchy : (1) Public topology (48 bits), (2) Site topology (16 bits) and (3) Interface Identifier (64 bits). The public topology is the collection of providers and exchange points. The Site topology is local to a specific site or organization. It is used by an individual organization to create its own local addressing hierarchy and to identify subnets. Interface identifiers identify interfaces on links.



**Figure2** IPv6 Aggregatable Global Unicast Address Format



**Figure 3** Modified IPv6 Unicast Address

For performance concerns, we propose to define a *Mobility bit* within the Site topology field of the IPv6 address format (see Figure 3). This bit, which is only meaningful within a site, is used by the site interconnection routers to demultiplex packets addressed to a mobile host from the packets addressed to a fixed host efficiently. The mobility bit of a host is set to 1 in mobile hosts' addresses and set to 0 in fixed hosts' addresses. By examining this bit, the site interconnection routers can instantly know if the incoming packet should be routed internally by the standard routing protocol or the local MISP. As a result, the packets addressed to fixed hosts do not suffer from the routers' MISP processing. The mobility bit is not a requirement. It is just a suggestion to speed packets' processing at routers. Note that the mobility bit does not require to be deployed in every sites and does not affect the routing of packets on the backbone since it is only meaningful and used within a site.



## 5. COMPARISON AND EVALUATION

In this section we compare the performance of our proposal and of Mobile IPv6. When comparing the performance of different mobility management schemes, several factors have to be taken into consideration. Among these factors, three are particularly important (Myles,93): (1) The *scalability* property of the schemes, i.e. how do the schemes behave as the network grows and the number of mobile hosts increases. (2) The *routing performance* of the schemes, i.e. what is the extra latency introduced by each of the schemes. (3) The *transition performance* of the schemes, i.e. how fast are the transition phases performed.

### 5.1 Routing and Transition Performance

The routing and transition performances of both schemes are quite similar. The routing is optimum, packets follow the shortest path from the correspondent nodes to the mobile host, and handoffs are performed locally in both proposals. In fact, in our proposal, local handoffs are managed within the site. In Mobile IPv6, while location updates have to cross the whole Internet to reach the mobile host correspondent nodes, a mechanism is provided to smooth out transitions. After switching to a new default router, a mobile node may send a Binding Update to its previous default router, asking him to redirect all incoming packets to its new Care-of Address.

### 5.2 Scalability Performance

The main performance difference between the compared approaches resides in their scalability property. The scalability property of a protocol can be evaluated in terms of *its overhead growth on the Internet* with the size of the Internet, the number of mobile hosts and the number of correspondent nodes. This overhead can be evaluated by comparing, for each proposal, their memory requirements and their signaling load, i.e. the bandwidth used by the control messages, such as the Binding Updates.

#### 5.2.1 Memory Requirement

We evaluate, in this Section, the memory requirement of each proposal.

Mobile IPv6 requires that (1) each mobile node maintains a list of its correspondent nodes and (2) each correspondent node maintains a binding per mobile host it is communicating with. The corresponding memory requirement,  $Mem_{MIP}$ , can therefore be evaluated as follow:

$$Mem_{MIP} = 2 \times \#MH \times \#CH \times Size_{entry}, \quad (1)$$

Where  $\#MH$  is the average number of mobile hosts on the Internet,  $\#CH$  is the average number of correspondent hosts of each mobile host and  $Size_{entry}$ , the size of the binding that a CH must maintain per MH, and a MH must maintain per CH.

Our hierarchical proposal requires that (1) each interconnection router of a site maintains a binding per mobile currently visiting the site, (2) each mobile node maintains a list of its correspondent nodes and (3) each correspondent node maintains a binding per mobile host, it is communicating with, that is not roaming within its home site. The corresponding memory requirement of our approach,  $Mem_{HMIP}$ , can therefore be evaluated as follows:

$$Mem_{HMIP} = [(1 + \gamma)\#CH + \#Routers] \times \#MH \times Size_{entry} \quad (2)$$

Where  $\#Routers$  is the average number of interconnection routers of a site in the Internet and  $\gamma$ , the percentage of the non-local mobility. According to [Kirb95],  $\gamma = 0.31$  therefore,

$$Mem_{HMIP} = [(1.31)\#CH + \#Routers] \times \#MH \times Size_{entry} \quad (3)$$

The gain (or loss) of our approach over the Mobile IP approach is defined as:

$$G_{Mem,AV} = (Mem_{MIP} - Mem_{ABA}) / Mem_{MIP} \quad \text{Or,} \quad (4)$$

$$G_{Mem,AV} = 0.345 - \#Routers / (2 \times \#CH), \quad (5)$$

This gain is displayed in Figure 4 as a fonction of  $(\#CH/\#Routers)$ .

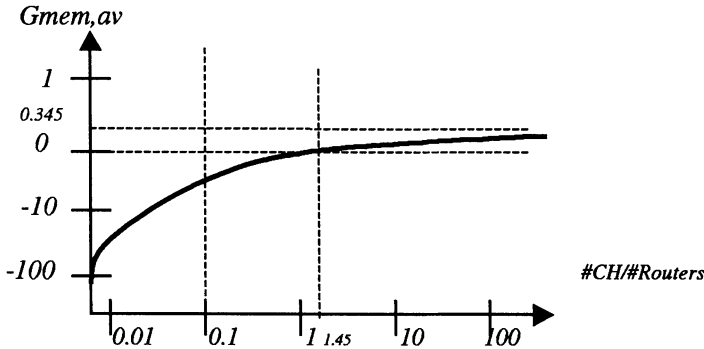


Figure 4 Memory Gain

These results show that our approach's memory requirement is lower than the Mobile IP one if  $\#CH$  is larger than a threshold,  $T$ , equal to  $1.45 \times \#Routers$ .

### 5.2.2 Signaling Load Overhead on the Internet (Backbone)

Both mobility management proposals use Binding Updates to set up states in the routers and/or the end-hosts. This signaling has a cost in terms of bandwidth

utilization on the Internet. In this section, we compare the signaling load introduced by both proposals on the Internet (backbone). We evaluate, for each of these schemes, the aggregated signaling load bandwidth consumed on the Internet. This aggregated bandwidth is independent of the number of nodes that the Binding Updates have to cross until their destinations, but rather corresponds to the signaling bandwidth on one link. To simplify, we do not consider the *Acknowledgement* messages that can sometimes be sent in response of Binding Updates. We do not compare the local signaling load because they are comparable for both schemes and because we argue that local resource is not the most critical. In this evaluation, we differentiate three types of mobility: *the local mobility of a host within its home site, the local mobility of a host within a foreign site, and the inter-site mobility of a host*. We then evaluate the average signaling load over these three mobility patterns.

### ***Binding Update Emission Frequency***

The signaling load of a scheme depends directly on the Binding Update Emission Frequency. According to (Perkins, 96b), a Binding Update is sent periodically to refresh the corresponding cache entries, and anytime a mobile host change its point-of attachment. The emission frequency of a Binding Update,  $freq$ , is therefore dependant on the mobility pattern of a host,  $freq_{MOV}$  and the refresh frequency,  $freq_{REF}$ . It is defined as follows:

$$\text{If } freq_{REF} > freq_{MOV} \quad \text{if } freq_{REF} > freq_{MOV} \quad (6)$$

$$Freq = a \times freq_{MOV} \quad \text{if } freq_{MOV} > freq_{REF} \quad (7)$$

With

$$a = \lceil freq_{REF} / freq_{MOV} \rceil \quad a' = \lceil freq_{MOV} / freq_{REF} \rceil$$

### ***Local Mobility within the Home Site***

When a mobile host, using Mobile IPv6, is moving within its Home site, it sends a Binding Update to each of its correspondent nodes and to its home agent at a frequency of  $freq$ . If our hierarchical proposal is used, no Binding has to be sent at all.

As a result, the signaling bandwidths respectively generated by Mobile IP,  $BW_{SIG\_MIP,home}(t)$  and by our proposal,  $BW_{SIG\_HMIP,home}(t)$ , when a MH is roaming within its home site, are defined as follows:

$$BW_{SIG\_MIP,home}(t) = Size_{BU} \times freq \times \#CH, \quad (8)$$

$$BW_{SIG\_HMIP,home}(t) = 0. \quad (9)$$

where  $Size_{BU}$ <sup>3</sup> is the size of a Binding Update and  $\#CH$  is the number of correspondent hosts that are not in the home site.

### ***Local Mobility within a Foreign Site***

When a mobile host, using Mobile IPv6, is moving within a foreign site, it sends a Binding Update to each of its correspondent nodes and to its home agent at a frequency equal to  $freq$ . If our proposal is used, the mobile host only sends a Binding Update to each of its correspondent nodes and to its home agent at a frequency equal to the refresh frequency,  $freq_{REF}$ <sup>4</sup>. **As a result**,  $BW_{SIG\_MIP,foreign}$  and  $BW_{SIG\_HMIP,foreign}$  **are defined as follows**:

$$BW_{SIG\_MIP,foreign} = Size_{BU} \times freq \times (\#CH + 1) \quad (10)$$

$$BW_{SIG\_HMIP,foreign} = Size_{BU} \times freq_{REF} \times (\#CH + 1) \quad (11)$$

### ***Inter-Site Mobility***

The signaling bandwidth introduced on the Internet when a mobile node is transiting from one site to another is the same in both schemes. For each of these schemes, the mobile sends a Binding Update to its home agent, distant correspondent hosts and to the correspondent hosts that were in its previous site. Therefore,  $BW_{SIG,transit}$  is defined as follows:

$$BW_{SIG,transit} = Size_{BU} \times (\#CH + \#ch + 1) \quad (12)$$

Where  $\#ch$  is the number of correspondent hosts that are located in the previous site.

### ***Analysis of the Results***

In this section, we evaluate, for each of the mobility pattern, the gain achieved by our proposal over Mobile IPv6,  $G$ . We note  $G_{home}$  the gain when the host is moving within its home site,  $G_{foreign}$  the gain when the host is moving within a foreign site, and  $G_{transit}$  the gain when the host is transiting from one site to another.  $G_y$  (with  $Y = home \text{ or } foreign$ ), and  $G_{transit}$  are defined as follows:

$$G_Y = (BW_{SIG\_MIP,Y} - BW_{SIG\_HMIP,Y}) / BW_{SIG\_MIP,Y} \quad (13)$$

$$G_{transit} = (BW_{SIG\_MIP,transit} - BW_{SIG\_HMIP,transit}) / BW_{SIG\_MIP,transit} \quad (14)$$

We also evaluate the average gains,  $G_{AV}$  over the three mobility patterns, by using the result established in (Kirk,95) that 69% of a host's mobility is local.  $G_{AV}$  is defined as follows:

---

<sup>3</sup> The size of a Binding Update is equal to the size of an IPv6 header (40 bytes) + the size of a Binding Update Extension Header (28 bytes), so 68 bytes. A Binding Update can however be smaller if it is sends with some payload.

<sup>4</sup> The Binding Updates sent to the local correspondent hosts do not cross the Internet

$$G_{AV} = 0.69 \times G_{home} + 0.31 \times (\alpha \times G_{foreign} + \beta \times G_{transit}) \quad (15)$$

where  $\alpha + \beta = 1$ ,  $\alpha = (N-1)/N$  and  $\beta = 1/N$ ,  $N$  being the average number of different points-of attachment of a mobile host within a site.

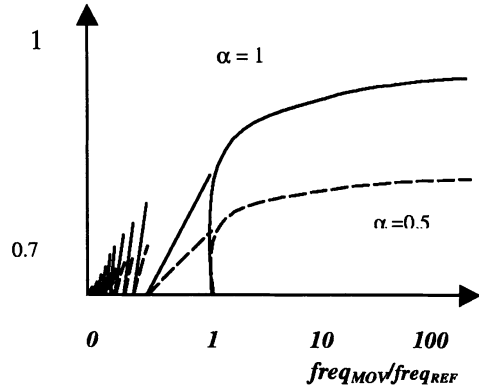
$\alpha$  and  $\beta$  characterizes the mobility pattern of a user outside of its home site.  $\alpha$  defines the intra-site versus inter-site moves ratio of the mobile hosts. A large  $\beta$  means that the user is frequently changing sites. A large  $\alpha$  means that the user is mainly roaming within a site and barely changes sites. For example, a  $\alpha$  of 0.9 means that the mobile host changes, in average, 10 times its point-of attachment within a site before moving to another site.

$\alpha$  and  $\beta$  can be written as functions of  $T$  and  $freq_{MOV}$ :

$$\beta = 1 / (T \times freq_{MOV}) \quad (16)$$

The achieved gains computed with the previous results are presented in the following table.

$G_{home}$	1.0
$G_{foreign}$	$(freq - freq_{REF}) / freq$
$G_{transit}$	0
Average Gain	$0.69 + 0.31 \cdot \alpha \times (freq - freq_{REF}) / freq$



**Table 1** Signaling Load Gains

**Fig. 5** Average Signaling Load Gain

These results show that our proposal's average signaling load on the Internet is at least 69 % lower than the signaling load generated by Mobile IP.

According to equations (6) and (7),  $f = (freq - freq_{REF}) / freq$  is defined as follows:

$$f = (a - 1) / a \quad \text{if } freq_{MOV} / freq_{REF} \quad (17)$$

$$f = (a' - freq_{REF} / freq_{MOV}) / a' \quad \text{if } freq_{REF} / freq_{MOV} \quad (18)$$

Figure 5 shows the average gain  $G_{AV}$  as a function of  $freq_{MOV} / freq_{REF}$  for  $\alpha$  equal to  $1/2$  and 1. This figure shows that the gain of our approach over Mobile IP is always larger than 69% and gets larger when a mobile host has a high mobility frequency,  $freq_{MOV}$ , and is mainly roaming within a site ( $\alpha$  is close to 1.0). The peaks that

appear when  $freq_{MOV}$  is smaller than  $freq_{REF}$  exhibit a synchronization problem of Mobile IPv6. In fact, when  $freq_{MOV}$  is smaller and is not a multiple of  $freq_{REF}$ , 2 consecutive Binding Updates are sent: one to refresh the cache entries of the correspondent hosts followed by the Binding Update sent when the mobile host changes its point-of attachment. This problem does not exist with our approach since the second binding is not sent on the Internet but is confined inside the site.

## 6. DISCUSSIONS AND CONCLUSIONS

This paper presents a mobility management architecture for the Internet that is hierarchical and flexible. The proposed scheme, which is fully compatible with the IETF solution, differentiates the inter-site mobility management from the intra-site mobility management. The hosts' local mobility is handled by a local, possibly customized, protocol while the global mobility, i.e. across sites, is handled by Mobile IPv6.

When a mobile host is roaming within its home site its mobility is fully hidden from its external correspondent nodes that see the mobile host as a regular fixed host. This property could be achieved with "regular" Mobile IPv6 by using the triangular routing mode (i.e. the mobile host does not send any binding update to its correspondent nodes). However in this case, and as opposed to our proposal, all packets addressed to a mobile host are first delivered to its Home Agent and then forwarded to the current mobile host's care-of address. These indirections can drastically increase the latency and increase the network load if the home site of the mobile host is large and the mobile host is far away from its home agent. *Our proposal proposes to use several home agents per mobile host.* One which is located on the mobile host's home subnet, as in the regular Mobile IPv6 protocol, to handle local communications (between the mobile host and correspondent hosts of the site) and the others located on the site interconnection' routers to handle external communications (between the mobile host and correspondent hosts outside the mobile host's home site).

When a mobile host is roaming within a foreign site, its local mobility (i.e. within this site), is hidden from its correspondent hosts. These hosts are aware that the mobile host is visiting the foreign site but are unaware of its local moves. This level hierarchy is not provided with the current IETF Mobile IPv6 proposal, which requires that a correspondent host be aware of all mobile host's moves. Note that Mobile IPv4, which is the protocol that manages mobility in IPv4, defines the concept of *Foreign Agents*. A foreign agent is an agent a mobile host may register with when it is visiting a foreign network. Packets addressed to the mobile host are then delivered to its foreign agent and forwarded to the mobile host. Foreign agents have originally been defined to limit the constraint of mobility on the short

IPv4 address space<sup>5</sup>, but (Caceres,96) shows that they can also be very useful in defining a hierarchical mobility management scheme. However foreign agents have not be maintained in Mobile IPv6, since IPv6 provides a much larger address space than IPv4. We argue that foreign agents should be reconsidered and adapted to Mobile IPv6 to define an hierarchical scheme. This paper proposes to deploy Mobile IPv6 foreign agents in the site interconnection's routers to hide mobile host's local moves from their correspondent nodes. These agents provide functions that are very similar than the home agents ones.

As shown in this paper, using a hierarchical mobility management scheme reduces the mobility management signaling load. In fact, we show that the signaling load generated by our proposal is more than *69% lower* than the Mobile IPv6 one. When a mobile host is roaming within its home site, no binding update has to be sent over the Internet. Beside from the Internet resource saving, eliminating the signaling has several other advantages. First, it reduces the risk of attacks and tracking of a mobile host. It also eliminates the need of authentication and encryption of the Binding Updates and the associated difficult issue of the keys distribution over the Internet. Two, eliminating the signaling allows to provide mobility management support to sites which are connected to the Internet with a unidirectional and/or an asymmetrical link, such as a satellite link. Three, eliminating the signaling load is important for scalability reasons. Mobile IPv6 requires that each host maintains one entry per mobile host it is communicating with. This requirement can be overwhelming for big servers, such as Web servers, that must maintain one entries for each of its mobile clients. By handing locally the moves of mobile hosts within their home site, we reduce the number of mobile hosts on the Internet and consequently the number of entries that each correspondent host should maintain.

## 7. REFERENCES

- Hari Balakrisnan, Srinivasan Seshan and Randy H.Katz (1995). Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks. *Proceedings of the first Annual International Conference on Mobile Computing and Networking (Mobicom'95)*, Berkeley, California, USA.
- Pravin Bhagwat, Satish Tripathi and Charles Perkins (1995). Network Layer Mobility: An Architecture and Survey. *Technical report CS-TR-3570*, University of Maryland.

---

<sup>5</sup> In Mobile IPv4, a mobile host does not get a new care-of address but borrows the address of its foreign agent

- Ramón Cáceres, Venkata N. Padmanabhan (1996). Fast and Scalable Handoffs for Wireless Internetworks. In *Proceedings of the Second Annual International Conference on Mobile Computing and Networking*, Rye, New York, USA.
- Claude Castelluccia (1998). A Hierarchical Mobility Management Scheme for IPv6. *Proceedings of the Third IEEE Symposium on Computers and Communications*, Athens, Greece.
- Mike O'Dell. GSE (1997) – *An Alternate Addressing Architecture for IPv6*. Draft-ietf-ipngwg-gseaddr-00.txt.
- R. Hinden, M O'Dell and S. Deering (1998). *An Aggregatable Global Unicast Address Format*. Internet Draft.
- G. Kirby (1995). Locating the User, *Communication International*.
- Andrew Myles and David Skellern (1993), Comparing Four IP Based Mobile node Protocols. In *Proceedings of the 4<sup>th</sup> Joint European Networking Conference*, Thondheim, Norway, pp. 191-196.
- Charles E. Perkins, editor (1995). IP Mobility Support, *Request For Comment*, RFC2002.
- Charles E. Perkins and David B. Johnson (1996), Mobility Support in IPv6. In *Proceedings of the Second Annual International Conference on Mobile Computing and Networking*, Rye, New York, USA.
- K. Sklower (1991), A Tree-Based Packet Routing Table for Berkeley Unix. In *Proceedings of the 1991 Winter USENIX Technical Conference*.
- Fumio Teraoka et al. (1992), Design Implementation and Evaluation of Virtual Protocol. In *Proceedings of the 12<sup>th</sup> International Conference On Distributed Computing Systems*, pp.170-177.

## 8. BIOGRAPHY

Claude Castelluccia is a research scientist at INRIA Rhone-Alpes, France. He holds a MSEE from Florida Atlantic University, USA, and a PhD from INRIA Sophia-Antipolis, France. His research interests includes mobile internetworking, protocol design and signal processing.



# Active libraries: A flexible strategy for active networks

*David C. Lee and Scott F. Midkiff*

*Bradley Department of Electrical and Computer Engineering*

*Virginia Polytechnic Institute and State University*

*Blacksburg, Virginia, USA 24061-0111*

*Phone: +1 (540) 231-2295*

*Fax: +1 (540) 231-3362*

*Email: dlee@vt.edu and midkiff@vt.edu*

## **Abstract**

Active networks are a new area of research and as such there are many different ideas as to what an active network should be. One possible architecture is described in this paper along with experimental results that characterize the performance and verify the operation of a fundamental component of the architecture. This component is an active library resolution service that allows active programs to find and load active libraries from the network. This strategy is evaluated and reviewed in detail. The results indicate that the research prototype, which only investigates the issue of where and how to obtain arbitrary libraries from the network, does work and should be scalable. The proposed architecture relies on a novel conceptualization of what minimum functionality should be in an extensible operating system and how systems could be built in an active network. The view is that extensible operating systems and mobile code allow functional components of the operating system and user applications to be obtained from the network. Thus, the overarching goal is a dynamic system in which system modules, such as file systems or network protocols, are loaded into and unloaded from the kernel on demand and the modules are obtained transparently from the network for active programs that need them.

## **Keywords**

**Active libraries, network protocols, active networks, operating systems**

## 1 INTRODUCTION

The evolution of the modern Internet is a process that has taken decades and has resulted in a robust and flexible service to an ever-growing number of users. This rapid increase in usage and addition of service requirements is straining the Internet and forcing the development of a new protocol for the network, Internet Protocol version 6 (IPv6) (Lee, *et al.*, 1998). As IPv6 development has also taken a number of years and will continue to take a number of years, various “glue” technologies have been and are being used in the interim until there is a pure IPv6 network. The ubiquitous World-Wide Web had been around a number of years before it became popular. These two cases illustrate an important problem in protocol standardization and deployment – it takes a significant amount of time to develop a protocol and to have significant wide-spread adoption of a protocol to make it truly useful. This problem is especially troublesome for protocols that operate at the network or transport layer; users must have a supported platform and/or must be able to get their service providers to support the service. For example, IP multicast service deployment has been resisted by many service providers. One proposed concept for reducing protocol development, deployment, and acceptance time is the active network (DARPA, 1998a and Tennenhouse, *et al.*, 1997). An active network is a network that is dynamically configurable and allows for “rapid injection” of new protocols. This is done by not standardizing on how bits are transferred across the network, but rather on uniform computational models for protocol processing.

This paper presents an architecture that supports the dynamic nature of an active network by allowing active programs to find and load arbitrary active libraries. It discusses the operation and implementation of a research prototype that was used to verify the core component of the architecture, the active library resolution service. The paper also presents the evaluation of this prototype which shows that it works and works well.

## 2 ACTIVE NETWORK RESEARCH

Tennenhouse, *et al.* (1997), provide an excellent summary of current active network research. Important active network concepts used in this work are briefly reviewed. In the modern network, a packet is the delivery unit that transfers data from one point to another. The concept behind active networks is to make the packet “smarter” by inserting code as data. This code is executed at intermediate switching nodes and allows custom computations to be performed on arbitrary packets. Clearly, there are a number of problems that arise and these are discussed below in relation to the enabling technologies.

A subtle change in how code-carrying packets are viewed can result in a powerful abstraction since, in this case, the packet is not a vehicle for carrying code but it is the code itself. This new vehicle is commonly called a capsule (Tennenhouse and

Wetherall, 1996). Capsules are self-contained programs that instruct programmable switches on how to process themselves and may leave behind persistent code to help process other capsules.

### 3 ENABLING AND RELATED TECHNOLOGIES

Mobile agents (Nwana, 1996 and IBM, 1997), a type of intelligent agents, are very similar in concept to an active network. Mobile agents are software that can be executed on arbitrary nodes in the network and can forward themselves through the network. The major difference between mobile agents and active networks is that mobile agents are at the application layer and active networks are generally at the network layer.

Key enabling technologies for an active network include extensible operating systems (Engler, *et al.*, 1995, Shapiro, *et al.*, 1997, and Bershad, *et al.*, 1995) and mobile code (Thorn, 1997, Yemini and da Silva, 1996, and Wahbe, *et al.*, 1993). Extensible operating systems take the micro-kernel approach one step further – they separate resource management from resource protection (Engler, *et al.*, 1995). Thus, the extensible operating system can dynamically load different modules at run-time, such as memory managers or file systems. A run-time user-customizable kernel has a number of problems such as efficient inter-module communications and secure execution. One clearly does not want a poorly behaved module to terminate the entire operating system.

Mobile code allows platform-independent, or portable, software to be developed. Mobile code technology, such as Java (Gosling, *et al.*, 1996) and NetScript (Yemini and da Silva, 1996), comes in two variants, interpreted and dynamically compiled. Interpreted code is executed by software that performs the function described by the source code or scripting language. An alternative approach to interpretation is to run a machine-neutral bytecode in a virtual machine environment. Dynamically compiled code takes the portable source code and compiles it so that a native machine-code binary can be used. Clearly, dynamic compilation is faster than interpretation; however, there are numerous code-safety issues that must be resolved (Wahbe, *et al.*, 1993 and Keppel, *et al.*, 1991). Both mobile code execution mechanisms should be available in a truly flexible active network environment. The Liquid Software project shows that it is possible to compile simple Java code to native code on a 200-MHz machine in the time it takes to receive the code on a 10 Mbps network connection (Hartman, *et al.*, 1996).

Extensible operating systems can be combined with mobile code to allow truly dynamic operating environments. This combination allows a vendor to write one application that runs on any hardware platform that incorporates the mobile code and extensible operating systems in a standard way. Obviously, much research and standardization remains to be performed in order to efficiently, reliably, and securely perform this task. This research focuses on one extension to, or

modification of, this model. If the application needs a module, such as a database manager, that is not currently on the system then the application should be able to resolve it from the network. If the application was properly debugged and written, the module that is required must exist somewhere and there is a high probability that it can be found on the network.

#### 4 ACTIVE NETWORK OPERATING SYSTEMS

The system model proposed by this work is a synthesis of the active network capsule concept, mobile agent operation, the extensible nature of research operating systems, and use of mobile code. The premise is that there is an operating environment, called the active network operating system (ANOS), that is run-time extensible and supports mobile code. It is divided into three privilege levels, the kernel space, the active handler space, and the user application space, as shown in Figure 1. Operating system “personality modules” are referred to as active handlers. The kernel space only provides basic resource protection and access functions, such as access to a file and prevention of multiple simultaneous writes. The active handler space provides the application services that users require, including file systems, memory managers, network protocol stacks, etc. Thus, different users can select different file systems according to what best suits their need. The minimally required modules are an active code resolution module and a network interface. If the resolution module is robust enough, all other modules can be obtained from the network. The user space is akin to the traditional user space.

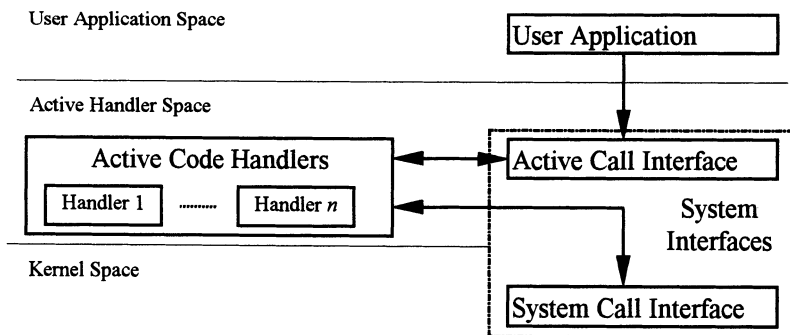


Figure 1 Block diagram of the ideal operating system at a node.

A brief discussion of the active network operating system architecture is given below. Details of the proposed architecture can be found in (Lee, 1998). There are two types of active code, one that exists within the operating system itself and the other that exists in user space. The user space code can be viewed as a mix of traditional applications and mobile agents. The handler code can be viewed as a

mix of traditional extensible operating system modules and active protocols sent across the network. The discussion focuses on handler code as opposed to user code.

## 4.1 System Bootstrap

Assume that a system is shipped with a minimal operating system of the core handler, a base protocol, and a resolution service. The base protocol provides network services for the resolution service and uses the bare memory protection and access features of the kernel. If a form of Dynamic Host Configuration Protocol (Dhrows, 1997) is embedded into the resolution service, then administrators at a site can ensure that all nodes are loaded in the same fashion. The only special modification that is required in the proposed resolution service is that a fixed sequence of resolution requests must be used. No other special protocol is required. Upon bootstrap, the node executes the initialization code which performs a set of resolution requests for a compiler, an interpreter, a memory manager, a file system, and so forth. Note that the interpreter and/or compiler must be able to support the native binary format of the machine in question and it may be preferable to include a base dynamic compiler in all active network operating systems. If so, then this compiler should allow itself to be replaced. Once the compiler is installed, all other modules can be made to run on the node in question.

## 4.1 Active Handler Resolution

The precise installation process for active handlers is still an open area of research. There are numerous communications and code safety issues that need to be addressed; however, run-time insertion of dynamic code has been proven to be possible within an Active Bridge (Alexander, *et al.*, 1997a). A model of how the resolution strategy operates from the network standpoint is provided later. The remainder of this section describes how code could be resolved and inserted into the operating system. The assumption is that there is a mechanism to query the network for active code, for the network to return information about the location and properties of active code, and for the network to return the code itself.

Figure 2 shows a block diagram of the resolve handler. Clearly some mechanism must be present to ensure that the code that is received is 1) authorized to be executed by the node, and 2) complete. Assuming that code is somehow received by the system, two levels of authorization checking are performed by the proposed system. The gatekeeper makes a simple high-level check to verify that the code format is supported and the capsule was received intact. The security service provides more detailed authentication mechanisms before the code is either directly loaded into the handler space, compiled into code that can be loaded into the handler space, or interpreted within the handler space. Code that is loaded into the handler space runs using special guard code (Wahbe, *et al.*, 1993) to ensure that

handler failure does not cause catastrophic failure for the operating system and to ensure that the handler does not try to override operating system protections.

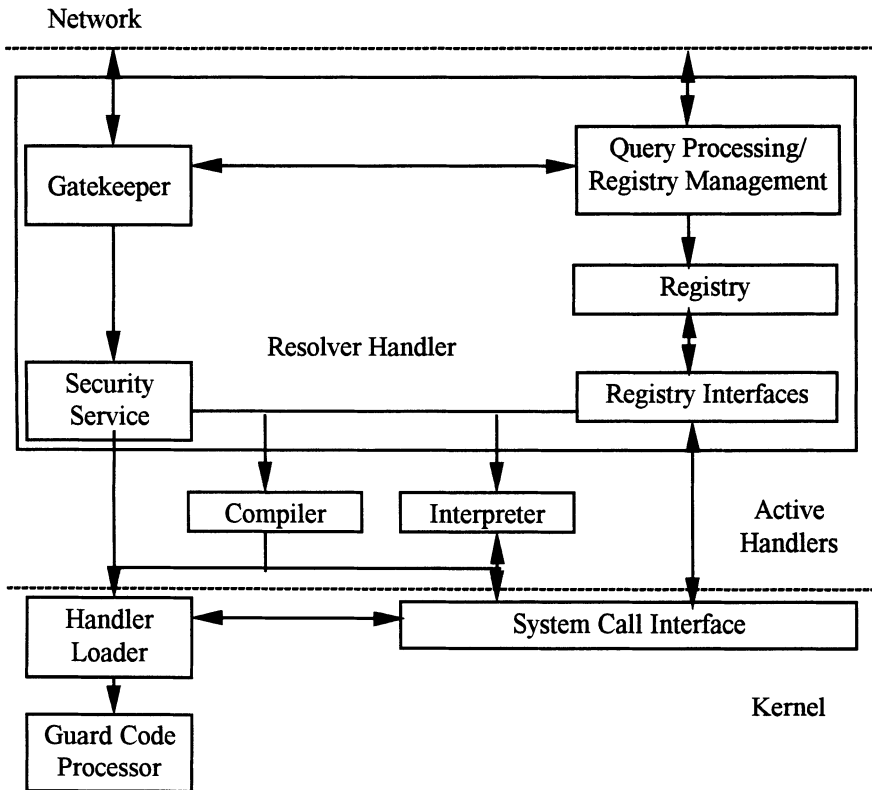


Figure 2 Block diagram of the resolver handler.

Code delivery, from the user space, in the user space, and from the network, may not be complete. For example, a user application may be run that requires a newer windowing system library than is present on the system. Without that library, the application will not run; however, if the code is active in nature then it can make a call to the resolution service which will search the network for the appropriate library. Once that library is found, it can be installed and the application can be run – all without human intervention. Two mechanisms can be used to ensure that the code is complete. The first mechanism is to embed, in special headers transferred along with the code, all the required libraries. The gatekeeper can use this information to verify that required libraries are present and if not present, can either obtain them for the active code or reject the active code. The second mechanism is the on-demand resolution and loading of active libraries. This can be performed by the guard code that is inserted into the active code or by the

interpreter. The former method allows for efficient, non-stop execution of the active code and the second method ensures that the system is robust.

## 5 ACTIVE CODE TRANSPORT AND RESOLUTION

The remainder of the paper discusses how code can be transported to and from a node and a proposed strategy for resolving libraries from the network. There are two methods that can be used to deliver active code, retrofitting existing protocols or creating a new internet protocol. Clearly, the later is not a preferred solution if the only driving reason to create a new internet protocol is to support active code. Thus, retrofitting IP is the popular solution (Wetherall and Tennenhouse, 1996 and Alexander, *et al.*, 1997b). The solution entails creating a new IP header option, the Active IP option (Wetherall and Tennenhouse, 1996) that can be used to inform a node that the code in the packet is a capsule. Nodes that are aware of the active network can process the code and nodes that are not aware of the active code will simply ignore the code. These retrofitting proposals provide a way to identify the program encoding that is used by the active code and authentication information about the active code. This is insufficient as other information will be required for a robust active network; however, the design of the Active IP option does not preclude this as it uses an encoding similar to Multipurpose Internet Mail Extensions (MIME) headers (Borenstein and Freed, 1996). Thus, if information such as distribution restrictions, copyright and usage cost information, revision history, required libraries, and so forth need to be attached to active code, it can be easily done.

The retrofitting proposals provide a simply delivery mechanism that does not easily allow active code to be retrieved from another node. One can make the argument that a capsule can be sent to the other node which can then retrieve the active code. If so, this would be a standard operation and this capsule can be viewed as part of a standard set of capsules that are available on any active node. The retrieval operation is an important requirement for a robust active library resolution strategy. The active library resolution strategy should be able to request code to be transferred or, upon receipt of a query and finding the library locally, a node should be able to send to code to the requester.

Now that the basic elements of the operating environment and code transfer mechanisms have been developed, the core service of active library resolution is discussed. Active libraries are active code that are used by other active programs. The libraries can be stand-alone programs or other libraries.

Assume the active code arrives at switch C and passes to switch B and then to host A and also assume that an active library is required, as shown in Figure 3. Ideally, host A would request the library first from switch B and then from switch C (Tennenhouse and Wetherall, 1996). However, it is possible that switch B and C can either fail or delete required libraries for whatever reason and host A will not

be able to execute the active code. If the active code is popular, then clearly the active library will be located somewhere on the network; especially considering that users within a group tend to use the same applications (Alexander, *et al.*, 1997a). To handle the situation that the active library may be found along the return path of the code or is available locally, the proposed service relies on the expanding-ring multicast search (Deering, 1991). This mechanism also allows queries to arbitrary and unknown nodes on the network. The expanding-ring multicast search first makes a query to all nodes that are on the same network as the requester, labeled search 1 in Figure 3. If a library is not found on the local network, then the search is repeated so that adjacent networks are included, labeled search 2. Again, if the library is not found, the search range is increased again. If proxy servers are used, then this model can be expanded to handle private networks and potentially provide query translation from different protocols. In addition, caching mechanisms can be used so that special active library servers can be present in the network.

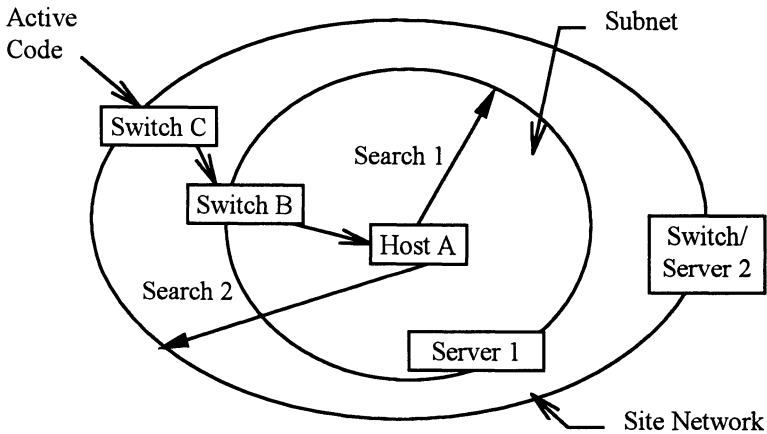


Figure 3 Active library resolution operational model.

## 6 PROTOTYPE IMPLEMENTATION AND EVALUATION

This research developed an experimental protocol to perform active library resolution and a related protocol to transport the active code. The transport protocol was developed to show the features that are required of a transport protocol for an active network. A prototype implementation was created to verify that the resolution and code execution strategy works. As it is impossible to measure the performance of a global service without wide-scale deployment and the later is clearly not possible, a separate simulation system was built to verify that the performance of the system is acceptable and scalable. The simulation



system was validated against test runs of the small-scale prototype. The remainder of this section discusses key features of the transport protocol and the library resolution protocol and presents the experimental results.

## 6.1 Active Code Transport and Resolution

As discussed earlier, active code transport can be performed by retrofitting existing protocols and should include MIME headers. MIME headers are text-based and are easily extensible. This work developed a test protocol to determine what type of headers are required of an active network and what type of protocol, or capsule, support is needed. A wide range of headers from naming the active code, application programming interfaces (APIs), authoring and copyright information, distribution restrictions, usage cost, and payment were investigated. The proposed use of MIME headers also introduce a dictionary-based compression scheme to reduce the overhead consumed by text headers.

The headers also provide the mechanism to search for a library. Many of the headers can be used as search targets for a query. The searches are performed by making regular expression comparisons and simple numerical tests, as appropriate. If an active program requires an active library, the name of the active code is will typically be used as the search target. Search constraints include verifying that the active code in question has the correct API and, perhaps, version or other information. The search targets and constraints are treated as a logical AND of all previous operations.

Clearly, in a dynamic Internet of the future, code will be treated as a commodity and it is unclear what the economic model will be. Thus, this header investigation must be considered preliminary but the results are that any active code transport mechanism should support active code delivery and retrieval and provide a mechanism to allow capsules at a source node to determine if the remote node properly processed the capsule. As these are common requirements, standard support must be provided. This research implemented a simple protocol called the Active Transport Protocol (ATP) to test these ideas and to provide support for the resolution service.

As noted earlier, the resolution strategy relies on multicast to perform the query. Considering a set of MIME headers are used as search targets and constraints, it is clear that the query will not fit into a single packet. As the API information is expected to be the major component of the search, twenty Linux header files were investigated to determine the likely typical size of an API search.

As shown in Tables 1 and 2, the average function name size is 11 bytes and the average number of parameters per function is three. The average number of composite data types per function is six. This number is obtained by dividing the composite type count of 115 by the 20 header files. The average length of a

composite data type name is ten bytes and the average number of variables used by a composite data type is four. About 16 functions and 12 composite data types exist in an average header file. Other analyses, reported in (Lee, 1998), confirmed that these numbers are reasonable. Because function parameters typically consist of well-known structures and names, such as the `C int` or `float` primitive data types, a high-level of compression can be achieved. Based on this analysis and the typical maximum transfer unit of 1,500 bytes, it is likely that most queries will be one to three packets in length.

Table 1 Header File Analysis Results, Part I

	<i>Count</i>	<i>Total</i>	<i>Function Name Size (Bytes)</i>		
			<i>Average</i>	<i>Low</i>	<i>High</i>
Function Count	310	3409	11.0	2	30
Composite Types	115	1065	9.3	2	16
Totals	425	4474	10.5	2	30

Table 2 Header File Analysis Results, Part II

	<i>Total (Bytes)</i>	<i>Number of Parameters</i>		
		<i>Average</i>	<i>Low</i>	<i>High</i>
Function Count	675	2.1	0	9
Composite Types	407	3.5	1	26
Totals	1082	2.6	0	26

Considering the small number of packets required, the Active Library Resolution Protocol (ALRP) was designed to support up to 64 packets to be transmitted, which allows an estimated 590 uncompressed search constraints to be included in 576-byte minimum sized IP packets. If a library requires more than 590 search constraints, it probably is not well written.

ALRP uses no error correction, but relies on the fact that in the expanding-ring search the packets are naturally retransmitted as the ring is expanded. This significantly reduces the complexity of the protocol and the traffic required to transmit error correction information.

## 6.2 Simulation Evaluation

The prototype system verified that active libraries can be resolved from the network and installed and executed dynamically. A modified *ping* (Regents, 1993) program was delivered from a source node to a destination node. This ping program required a simple checksum library and this requirement was embedded in the headers. The library was resolved from the network, installed, and the ping program successfully executed.

Table 3 Test Network Characteristics

<i>Case</i>	<i>Group Membership Size</i>	<i>Level</i>	<i>Distance</i>	<i>Number of Leaf Networks</i>
1	10	0	2	2
2	100	0	2	2
3	1 000	1	3	10
4	10 000	2	5	100
5	100 000	4	9	1000

Since system extensibility can be achieved through the addition of MIME headers, the other primary criteria for a global system is scalability. As noted earlier, the system must be simulated to evaluate performance. The simulator that was created used random hierarchical networks at various fixed levels of hierarchy.

Hierarchical networks closely match the topology of a multicast tree (Deering, 1991). A number of different cases were used to show scalability, as indicated in Table 3. The number of nodes in the network and the levels of the network were increased for each case. The system measured the transmission time and number of packets sent and also allowed for different link error rates to be set. Two different end-to-end error rates were calculated and used which were one and five percent. One test case used error rates from one to 75 percent. Because of the lack of error correction, the important factor for scalability was the distance away from the source. Scalability was measured in terms of linear resolution time and packet counts as the distance from requesting client to the designated server increased.

The simulation did not account for the effects of multicast routing, caching, different request sizes and implementations, and similar considerations. The simulation does consider networks of different sizes, different loss rates, variations in server and source locations, and variations in key factors. Bandwidth for all

links is assumed to be a uniform 10 Mbps. Delay and loss rates are assumed to be uniform across all links. The simulator was verified against the prototype for case 1.

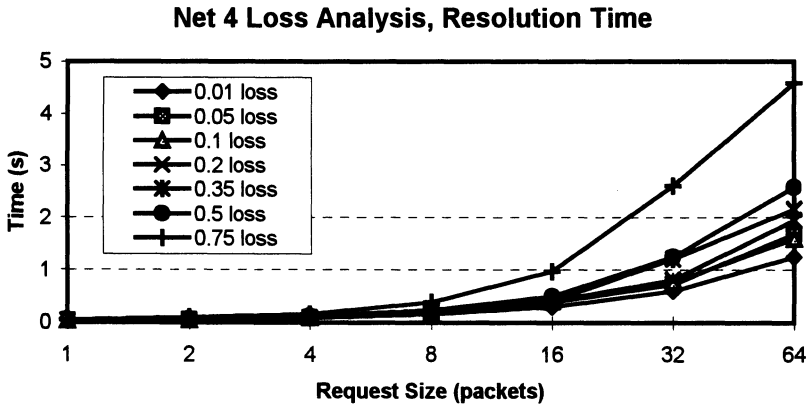


Figure 4 ALRP loss performance analysis for resolution time.

Figure 4 shows the analysis of loss for case 4. As the request size and the error rate are increased, the resolution time shows a proportional increase. Considering the worst case of 64 packets in a query, there was only a four times increase in the resolution time and a 3.3 times increase in the number of packets transmitted for the 75 percent loss rate compared to the situation with a one percent loss rate. For the typical case of one to three packets, there is a negligible difference in performance.

Figure 5 shows the analysis of scalability for the one percent and five percent error loss situations. This plot gives the resolution time as a function of the number of links and the results presented are an average of the normalized data set. Each data set was normalized to the case where the request size was one packet. The average of the four cases are plotted. For each loss rate, a linear “best-fit” curve is generated and its equation and  $R^2$  value are presented. The closer the  $R^2$  value is to unity, the more accurate the linear regression. Figure 5 shows a linear correlation for both the one percent and five percent resolution time cases. The slope for both curves is significantly less than one. Also, both equations are almost the same. This indicates that resolution time scales well, regardless of error rates, and that results for larger networks can be determined by the two similar equations.

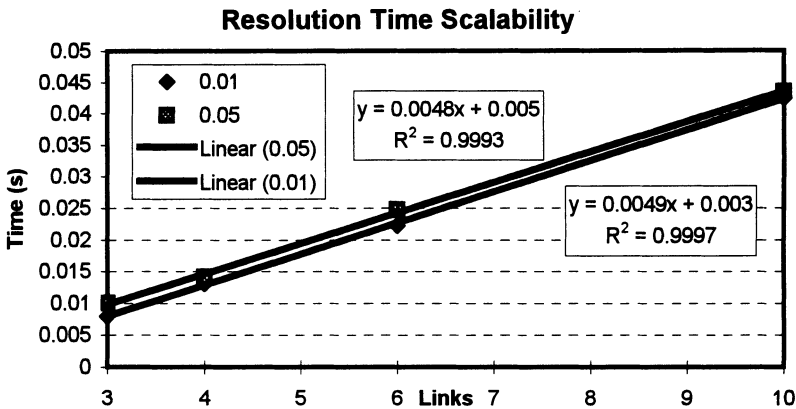


Figure 5 ALRP resolution time scalability.

For transmitted packet counts (not shown), the one percent error case has a slope of 1.3 and the five percent error case has a slope of 19. This indicates that low loss rates scale significantly better than higher loss rates with respect to the transmitted packet counts. Thus, the impact on the network is insignificant for the one percent case and moderate for the five percent case for moderately sized to large networks. From the above analysis, indications are that unrealistically huge networks will have a large number of retransmitted packets at five percent loss. Source to destination distances in modern internets generally are not much more than 15 hops. The significant difference in the two transmitted packet count equations indicates that higher error rates will lead to significantly more packets being retransmitted. Excessive numbers of packets will begin to degrade resolution time performance as network capacity is exceeded. Caching systems should reduce the average distance to something much less than 15 hops.

## 7 CONCLUSIONS

Active networks are an emerging area of research that relies on other emerging areas of research, most notably concepts from extensible operating system and mobile code. This work proposes a unique synthesis of these two areas by advocating a generalized paradigm of dynamic networks and nodal processing. A node consists of a uniform set of kernel interfaces and a service to dynamically obtain other system components from the network. A specific approach to the resolution service was discussed and an evaluation of this approach presented. The use of a flexible transport strategy combined with MIME headers allows for a powerful model of code resolution. Code is treated as an object and the entire network is treated as gigantic database. The scalability of the service is controlled by using expanding-ring multicast searches and caching systems. The

experimental prototype shows that this approach can work and the simulation results indicate that it can work well on a large scale.

Obviously, there is much research that must be performed before this model, or a variant of it, can be realized. There are significant issues in communications overhead, practical and scalable internal system architectures, standard access models, efficient and safe code execution, and unknown distribution and cost models. This does not even begin to cover the other challenges in active networks (DARPA, 1998b), such as network management and security.

## 8 REFERENCES

- Alexander, D.S., Shaw, M., Nettles, S.M., and Smith, J.M. (1997a) Active Bridging. *Computer Communication Review*, 27:4, 101-111.
- Alexander, D.S., Braden, B., Gunter, C.A., Jackson, A.W., Keromytis, A.D., Minden, G.J., and Wetherall, D. (1997b) Active Network Encapsulation Protocol (ANEP). *Internet Draft*. Work in progress. Available WWW: <http://www.cis.upenn.edu/~switchware/ANEP>
- Bershad, B.N., Savage, S., Pardyak, P., Sirer, E.G., Fiuczynski, M.E., Becker, D., Chamers, C., and Eggers, S. (1995) Extensibility, safety, and performance in the SPIN operating system. *Operating Systems Review*, 29:5, 267-84.
- Borenstein, N. and Freed, N. (1996) MIME (Multipurpose Internet Mail Extensions) Part one: Mechanisms for specifying and describing the format of Internet message bodies. *RFC 2045*.
- Deering, S.E. (1991) Multicast routing in a datagram internetwork. *Ph.D. Thesis*, Stanford University.
- Defense Advanced Research Projects Agency. (1998a) Mission. Available WWW: <http://www.darpa.mil/ito/research/anets/index.html>
- Defense Advanced Research Projects Agency. (1998b) Challenges. Available WWW: <http://www.darpa.mil/ito/research/anets/challenges.html>
- Droms, R. (1997) Dynamic host configuration protocol. *RFC 2131*.
- Engler, D.R., Kaashoek, M.F., and O'Toole, J. (1995) Exokernel: An operating system architecture for application-level resource management. *Operating Systems Review*, 29:5, 251-266.
- Gosling, J., Joy, B., and Steele, G. (1996) *The Java language specifications, Version 1.0*. Addison-Wesley.
- Hartman, J., Manber, U., Peterson, L., and Proebsting, T. (1996) Liquid software: A new paradigm for network systems. *Technical Report TR 96-11*, University of Arizona. Available FTP: <ftp://ftp.cs.arizona.edu/xkernel/Papers/tr96-11.ps>
- IBM, Inc. (1997) IBM Aglets workbench - home page. Available WWW: <http://www.trl.ibm.co.jp/aglets/>
- Keppel, D., Eggers, S.J., and Henry, R.R. (1991) A case for runtime code generation. *Technical Report UW-CSE-91-11-04*, Department of Computer Science and Engineering, University of Washington. Available FTP: <ftp://ftp.cs.washington.edu/tr/1991/11/UW-CSE-91-11-04.PS.Z>

- Lee, D.C., Lough, D.L., Midkiff, S.F., Davis, IV, N.J., and Benchoff, P.E. (1998) The next generation of the Internet: aspects of the Internet Protocol version 6. *IEEE Network*, 12:1, 28-33.
- Lee, D.C. (1998) Active library resolution in active networks. *Ph.D. Thesis*, Virginia Polytechnic Institute and State University.
- Nwana, H.S. (1996) Software Agents: An Overview. *Knowledge Engineering Review*, 11:3, 205-244.
- The Regents of the University of California (1993) *Ping*.
- Shapiro, J.S., Muir, S.J., Smith, J.M., and Farber, D.J. (1997) Operating system support for active networks. Available WWW: <http://www.cis.upenn.edu/~eros/devel/sigcomm97.300dpi.ps>
- Tennenhouse, D.L. and Wetherall, D.J. (1996) Towards an active network architecture. *Computer Communication Review*, 26:2, 5-18.
- Tennenhouse, D.L., Smith, J.M., Sincoskie, W.D., Wetherall, D.J., and Minden, G.J. (1997) A survey of active network research. *IEEE Communications Magazine*, 35:1, 80-86.
- Thorn, T. (1997) Programming languages for mobile code. *ACM Computing Surveys*, 29:3, 213-239.
- Wahbe, R., Lucco, S., Anderson, T.E., and Graham, S.L. (1993) Efficient software-based fault isolation. *Operating Systems Review*, 27:5, 203-216.
- Wetherall, D.J. and Tennenhouse, D.L. (1996) The ACTIVE IP Option. *Proceedings 7th ACM SIGOPS European Workshop*, Connemara, Ireland. Available WWW: <http://www.tns.lcs.mit.edu/publications/sigops96ws.html>
- Yemini, Y. and da Silva, S. (1996) Towards programmable networks. *Proceedings IFIP/IEEE International Workshop on Distributed Systems*. Available WWW: <http://www.cs.columbia.edu/~dasilva/content/netscript/pubs/dsom96.ps>

## 7 BIOGRAPHY

David C. Lee is a Visiting Assistant Professor in the Bradley Department of Electrical and Computer Engineering at Virginia Tech. He received the B.S. in Computer Engineering and the M.S. and Ph.D. in Electrical Engineering from Virginia Tech. His research interests include advanced network architectures, active networks, and run-time reconfigurable network hardware architectures.

Scott F. Midkiff is an Associate Professor in the Bradley Department of Electrical and Computer Engineering at Virginia Tech. He received the B.S.E. and Ph.D. degrees from Duke University and the M.S. from Stanford University, all in electrical engineering. His research interests include network protocols, wireless networks, and the application of network technology for education.

## **Part Six**

---

# **QoS in the Internet**



# End-to-end QoS Provisioning through Resource Adaptation

*D. G. Waddington and D. Hutchison  
Distributed Multimedia Research Group,  
Computing Department,  
Lancaster University,  
Lancaster LA1 4YR,  
UK  
e-mail: [dan, dh]@comp.lancs.ac.uk*

## Abstract

With the progression of multimedia middleware and guaranteed network services, developers are now presented with flexible frameworks for the development and deployment of distributed multimedia applications. New and more advanced applications are supporting end-to-end Quality of Service (QoS) guarantees through the configuration and management of distributed resources. As an effect of the sharing of network and end-system resources across multiple clients, coupled with their dynamically changing state, the general end-to-end availability of resources in a distributed environment is variable and potentially unpredictable. Thus, the provision of QoS constrained services in a distributed environment demands carefully controlled and co-ordinated management mechanisms. In this paper we discuss the requirements for QoS adaptation mechanisms and QoS-based distributed resource management, together with our approaches to QoS adaptation and policing, with issues concerning the incorporation of these mechanisms in our recently developed Distributed Resource Management Architecture (DRMA).

## Keywords

End-to-end QoS, Resource Management, QoS Adaptation

## 1 INTRODUCTION

The growth of general networked computing performance is being closely followed by the exploitation of this capability by QoS constrained applications. Many of

these applications, such as video conferencing and distributed collaborative environments, have dynamically changing and potentially unpredictable QoS requirements. This problem is exacerbated by the heterogeneous nature and varying capabilities of today's end-systems and global network infrastructures. As a result, conventional resource reservation and admission techniques cannot guarantee QoS without considerable over-booking and inefficient resource utilisation, something which is particularly undesirable in systems that are required to maintain high levels of resource sharing.

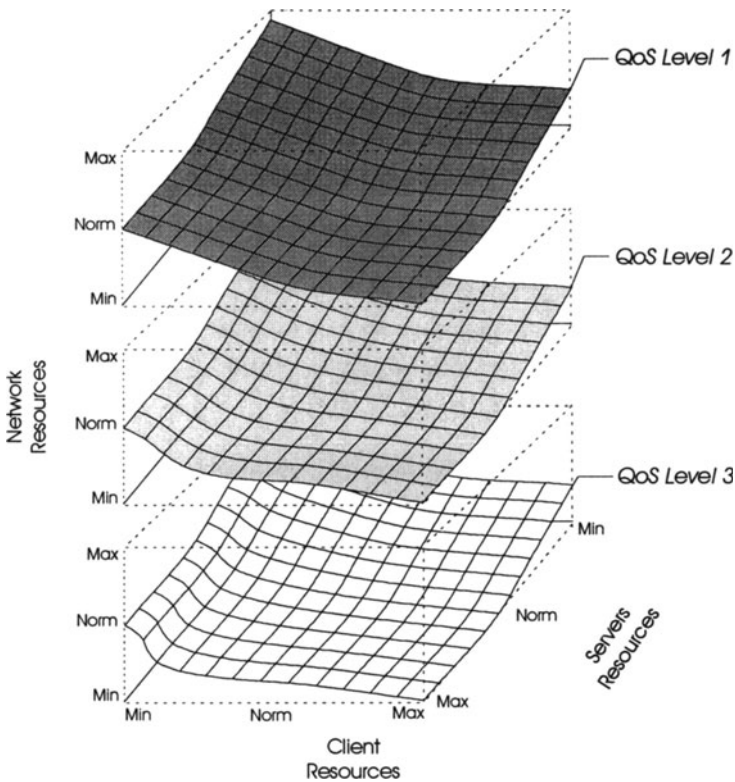
To address this problem, and avoid any adverse impact to the end-user, distributed applications and their infrastructures need to become adaptive. This means that either applications must tolerate fluctuations in resource availability or that the supporting infrastructure can itself mould, through distributed QoS management, to the dynamically changing requirements of the applications. Furthermore, certain applications, especially distributed multimedia applications with relatively demanding QoS requirements, simply cannot operate outside strict resource requirements, and therefore the need for QoS management arises. For example, Video on Demand (VoD) applications often incur varying resource requirements through changing counts of 'viewers' with different end-system capabilities (set-top boxes to high performance workstations). Furthermore, because of the heterogeneous nature of network connections and end-system resources, offering better guarantees and higher performance is often a infeasible and uneconomic solution. If QoS is not carefully managed, and resource utilisation scrupulously co-ordinated, then the desired levels service across the application are difficult to maintain. Other effects of poor QoS management include loss of synchronisation through delayed updates, deadlocks in shared resources, and failure of mission-critical applications through resource unavailability. The term 'QoS management' encompasses the maintenance of a required level of service through the co-ordinated configuration and control of end-to-end resources. Because of the obvious proliferation and acceptance of distributed object computing, and more importantly its usefulness in addressing the problem of QoS management in an open distributed system, proposals have already been made for object-based management frameworks (Dang, 1995)(ISO, 1997)(Waddington, 1997). However, up till now, much of the work has only addressed issues of QoS specification and the projection of useful QoS abstractions and services to the application programmer.

In this paper we discuss our approach to distributed QoS adaptation, furthering Lancaster's original work on QoS maintenance and adaptation in the QoS-A framework (Campbell, 1994). We now place more emphasis on the provision of QoS guarantees in distributed processing environments, whilst also addressing the problems of QoS-driven resource management and adaptation in the middleware infrastructure. In section 2 we discuss the general requirements for distributed resource management and the implications of supporting QoS constrained applications. Then, in section 3, we introduce the rationale for adaptive

QoS-based resource management, together with support for resource monitoring and QoS degradation predication, and its use in instigating adaptation processes. In section 4 we discuss some of the issues concerning the engineering and implementation of QoS adaptation into distributed multimedia middleware platforms, and in particular our recently developed Distributed Resource Management Architecture (DRMA) (Waddington, 1997). Finally in section 5 we present our conclusions.

## 2 QOS PROVISIONING THROUGH RESOURCE MANAGEMENT

In order to sustain the QoS requirements of a given continuous media application, all resources involved in the handling and processing of data from end-to-end, must be carefully co-ordinated and managed. The end-to-end QoS is some function of the resource utilisation of a distributed application, from client through network to server.



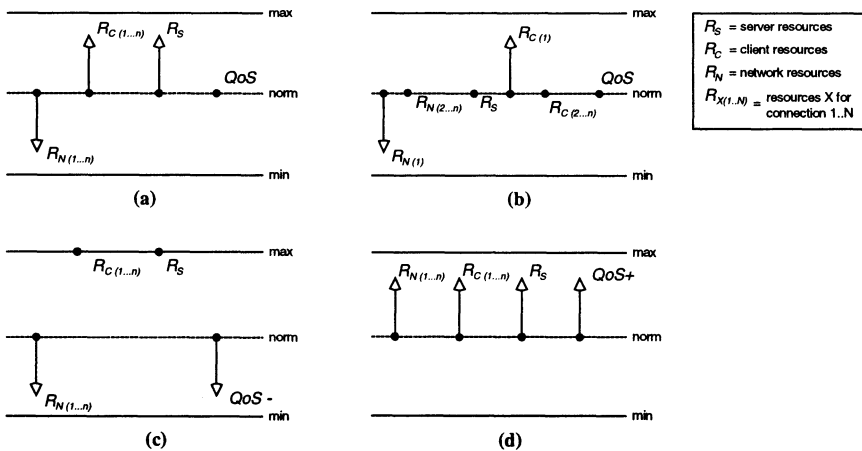
**Figure 1** Resource Capacity Regions for different levels of QoS

Any application which needs to provide a QoS-bound service must maintain its distributed resource utilisation within a finite space, which we call the *resource*

*capacity region*. Figure 1 illustrates this relationship between resource usage in the end-systems and the network. The region's surface represents the balance of resources required to sustain a particular end-to-end QoS; hence each level of service has associated a different capacity region. For each level of service, provided that an application maintains its resource utilisation within the defined region, QoS is sustained (note that each capacity region associated with a particular level of QoS is represented by its own graph).

A concept of capacity regions was proposed by Columbia University's work on meeting end-to-end QoS guarantees over high performance packet-switched networks. (Hyman, 1995) introduces the concept of a *schedulable region* which describes a finite space representing the number of calls of a given class a particular network link can support. This concept was later extended (Lazar, 1995) to incorporate capacity regions for end system resources, known as *multimedia capacity regions*, where the summation of the two capacities is used to determine end-to-end call admission.

If an application is unable to maintain its utilisation within the region, possibly due to the unavailability of resources, then degradation in the application QoS will occur. On the other hand, if an application's resource utilisation is not close to the surface of the capacity region, then resources are not being used optimally which may result in the degradation of QoS for other applications sharing the resources. Thus, distributed applications providing end-to-end QoS should strive to maintain resource utilisation as close as possible to the balance defined by the surface of the resource capacity region. So far, our discussion of the resource capacity regions has been limited to distributed applications based upon single client/network/server scenarios. However, the concept can be readily extended to multi-point communication scenarios, resulting in additional dimensions to the capacity graph (we are not limited by 3-dimensional space, as capacity regions are actually realised as non-visual multi-variable relationships).



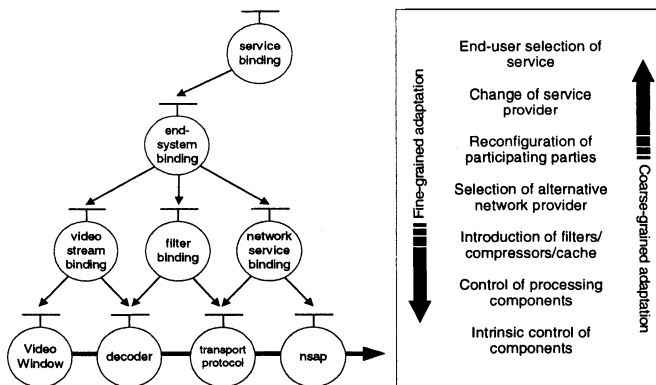
**Figure 2** Example Adaptation Scenarios in One-to-Many Relationships

*QoS adaptation* is the process of maintenance control, facilitated through alterations to either the balance and distribution of resources or to the application's level of service, on short time scales. Adaptation processes often occurs as a result of *QoS notifications*, usually emitted from QoS monitoring mechanisms, which indicate a change in the observed service affected through the availability of some element of the end-to-end resources. Notifications may indicate a imminent lack of resources and hence reduction in service quality (QoS degradation) or a failure to maintain service quality through a complete loss of resources (QoS failure). Whether degradation or actual failure occurs, QoS adaptation is required to either adjust the balance of resources to maintain graceful degradation, or recover service quality, or alternatively inform the end-user of the need to alter to a new level of service (change in level of service could entail dropping one or more media channels, or reducing information resolution). Ideally adjustments in resource balance, affected by the adaptation mechanisms, will satisfy the function of resource utilisation described by the surface of the resource capacity region. In doing so this is likely to involve one or more entities of the resource set (client, server or network) increasing their own resource utilisation to counteract deficiencies in the failing entity.

It is also important to realise that the rules of adaptation that we apply to simple client-network-server applications can be readily scaled to applications which communicate in one-to-many and many-to-many relationships. Figure 2 presents some example adaptation processes in a one-to-many environment. Figure 2(a) illustrates a loss in the network resources (for all connections) which is counteracted by an increase in client/server end-system resource utilisation (perhaps through the incorporation of compression techniques). In examining adaptation processes in multi-party relationships there are additional factors which must be considered, for instance the number of remote end-systems a particular server is communicating with. The example scenario described in figure 2(b) shows a drop in resources for a single connection, whilst other network resources remain stable. In this scenario it does not make sense to increase utilisation of the server and thus, of all the clients, in response to a single client failure. Therefore, an alternative action would be required, such as increasing the resource utilisation for the individual client (maybe by employing data reconstruction or forward error control techniques). If a particular set of resources fail and counteractions cannot be made, then a reduction in QoS is inevitable(see figure 2(c)). In such a scenario, the adaptation process is required to release other shared resources or inform the end-user of a need to change the level of service. Finally, figure 2(d) illustrates that in order to effect an increase in end-to-end QoS then an increase in all of the end-to-end resources is required.

## 2.1 Hierarchical QoS Management

From an engineering perspective, QoS management in a distributed system is a substantially complex task. An approach which has been proposed in the Distributed Resource Management Architecture (DRMA) (Waddington, 1997) is hierarchical QoS management. This technique breaks down the task of managing end-to-end resources by dividing the problem into a set of finer-grained point-to-point requirements which are structured as hierarchical bindings (see figure 3). By doing so, mapping and monitoring processes become distributed from end-to-end, enhancing scalability and avoiding problems of centralised control. The hierarchical management approach is also suited to the engineering of adaptation mechanisms. At the upper levels of the structure, adaptation mechanisms are responsible for coarse-grained actions, including reconfiguration and change of service. Descending towards the leaves of the hierarchy, adaptation mechanisms become finer-grained, usually in the form of atomic resource control. The processes responsible for the adaptation actions are maintained within binding components (bindings are simply communication abstractions between one or more potentially distributed object interfaces). Furthermore, each individual binding is responsible for maintaining, through monitoring and adaptation, the point-to-point QoS characteristics which are defined by its interfaces.



**Figure 3** Hierarchical QoS Adaptation

## 2.2 Core Distributed Resources

The term 'resource' is inherently vague. Resources can exist at varying levels of abstraction. For instance at the highest level the term could include an end-system or a network node. At a lower level, a resource could represent a physical device or some system wide shared resource such as a network connection or access to a physical disk. However, we believe that there is a finite set of resource, in the end-

system and the network, in terms of which all other resources can be described. Therefore we propose that, at least to begin with, we should concentrate on monitoring a limited set of resources and make adaptations accordingly.

**Table 1** Core Distributed Resources

<i>Area</i>	<i>Resource</i>	<i>Description</i>
End-System	CPU	Processor cycles
	Physical Disk Access	Disk I/O requests
	Memory	Paged and non-paged memory usage
	Cache	File and I/O caching
	Auxiliary Memory	Device buffers, video memory
	I/O Devices	Serial/parallel ports
	Peripherals	Video camera, microphone, etc.
	NSAPs	End-system access Ports
Network	Bandwidth	Traffic throughput
	Buffer Space	Router buffer requirements
	Switch/Router Ports	Network channels/paths

The core resources, as described in the above table, are shared by applications across the distributed environment. This finite set covers the majority of resources which are shared in a distributed system. By careful management and control of these, we can begin to prioritise resource utilisation and hence offer QoS-guarantees.

### 2.3 Resource Scheduling

In order to successfully share resources across distributed applications and, furthermore, offer sufficient guarantees on their availability (an obvious requirement for time critical continuous media applications), resources need to be scheduled. Scheduling is the process of determining the availability of resources at a particular instant in time. Through resource reservation, an application can request resources and in return the system can determine whether sufficient resources are available to service the given request.

Scheduling algorithms can only be effective if they are used in advance of the admission of the resource. However, the time scale between resource admission testing and admission is dependent upon the scheduling algorithm. Many algorithms, of varying complexity and usually focused at either the network or the end-system, have been suggested (Hyman, 1995)(Anderson, 1993). However, in scheduling resources for multimedia applications the use of exhaustive or statistical techniques is often inappropriate (applications such as VoD can be statically

analysed, but dynamically changing applications such as video conferencing cannot). It is suggested that simple heuristic scheduling techniques are preferable, offering a low processing overhead, and thus being more suited to resource requirements which vary in real-time. A basic model for resource reservation is offered by (Wolf, 1995). In this model, resources are *requested* by the application to the individual resource manager (i.e. network or end-system). The resource request describes the resource requirements of the application and the duration of the requirement. Provided that there are sufficient available resources, the resource manager returns a *confirmation*, otherwise the request is refused and a *failure* is returned. On successful completion of the negotiation phase, the client contacts the resource manager at the point when previously reserved resources are required. The *demand* is acknowledged, and the client can then use the resource for the duration of the reservation. This technique of reservation in advance does require that the duration of the reservation can be calculated *a priori* and that the resource usage is scheduled from the reservation request. There are techniques, such as partitioning, which can be used to couple with non-advance resource reservation systems; however we feel that the reservation in advance scheme is suitable for our purposes.

### 3 ADAPTIVE RESOURCE MANAGEMENT

Many of today's applications continuously adjust their resource requirements. This dynamic nature is particularly evident in distributed multimedia applications which demand strict levels of QoS. Furthermore, the problem is intensified in general purpose distributed environments because resources, in the network and the end-system, must be shared across multiple contending applications. Some operating systems, especially real-time systems, employ resource scheduling techniques (as previously discussed) in an attempt to alleviate the problem. Reservation and admission does allow a system to offer firm guarantees provided that the resource can be guaranteed, e.g. wireless communication bandwidth cannot always be guaranteed. However, many applications, such as distributed games, have varying resource requirements which cannot be predicted. One solution is to over-book resources and hope that the application does not demand resources beyond those made in the reservation. This is not an ideal solution as it is likely to result in the inefficient use of resources. So what is the rationale behind the use of resource reservation and admission techniques in a general purpose operating system at all? Why not simply rely directly upon monitoring and adaptation? In response, we suggest that compared with adaptation processes, resource reservation and admission processes are relatively lightweight and their processes demand little of the system. Furthermore, the majority of applications know their resource requirements *a priori*, and therefore are suited to a model of admission and reservation.



Coupled with the problem of varying client requirements, is the indeterminate nature of end-to-end resource availability. This difficulty is particularly evident within wide-area QoS guaranteed networks, which create an end-to-end connection through the concatenation of multiple hops. Because each hop in the connection continuously change state, QoS routing techniques must be employed in determining a suitable end-to-end route. As a result of the inability to determine the exact state of end-to-end resources, we propose the use of two-tiered 'loose' reservation and admission in order to maintain a best-effort management of resources. Loose resource reservation implies that approximate state metrics are used in admission and reservation, and that only statistical guarantees can be made. In the event that an applications resource requirements do change, then adaptation techniques are used to maintain QoS. Thus, the resulting system combines the benefits of resource reservation and admission, with the flexibility of monitoring and adaptation.

### 3.1 Monitoring and Prediction

Monitoring is the process of observing the utilisation of resources and/or QoS characteristics in the system. It is the responsibility of the monitoring process to observe events and provide messages indicating the occurrence of QoS contract violations. There are two approaches to monitoring, *intrusive* and *non-intrusive*. Intrusive monitoring means that the monitoring process takes periodic samples of resource availability and utilisation; this in turn means that resources are consumed by the monitoring process itself, an overhead which is sometimes unacceptable. Alternatively, monitoring processes can rely upon indications of events which are not within the bounds of an agreed QoS contract. For instance, a monitoring process may receive indications from a media decoder concerning the dropping of video frames. This approach means that the monitored data set is much smaller and often aperiodic, however, resources are only consumed by the monitoring process in the event of QoS degradation or failure (some researchers may argue that this would cause deadlocks since a system should not consume more resources at a point of QoS failure).

Many traditional QoS-based adaptive systems use indications of QoS failure to initiate adaptation actions. As a consequence any resource management and adaptation which is carried out by the system is, more often than not, readily noticeable to the end-user. To avoid such disjunction, it is suggested that we should attempt to predict the need for adaptation. Some sources of monitoring, such as the CPU usage of a video decoder, are often unpredictable<sup>1</sup>. You can see from the results in figure 4 that the utilisation of resources is often application and implementation dependent. The resource utilisation from the playback of a local

---

<sup>1</sup> The figures were taken from PerfMon on Windows NT 4.0 whilst decoding videos of comparable content through DirectShow 2.0 software decoders.

MPEG video is relatively periodic, however the playback of an Indeo video which can be considered to be a similar application, uses CPU resources much more sporadically. Furthermore, even if resource utilisation patterns can be identified, they are often too fine grained to be useful. Nevertheless, prediction techniques are useful for coarser grained trends in resource utilisation and/or QoS characteristics. In such cases we can introduce simple statistical prediction algorithms, such as extrapolation and regression, to make an estimation of the probability of a failure occurring. Prediction techniques do depend upon the resource being monitored and the process<sup>2</sup> variants which are causing the system to degrade. Variants are either *system-driven*, such as caching techniques, or *user-driven*, such as a new selection of service. User-driven variants are often stochastic processes (meaning that they are random and have a mean of zero), and are therefore very difficult to predict in the short term. However, system-driven variants often result in a relatively predictable pattern, allowing statistical prediction algorithms to be used to extrapolate future resource variations and hence enable adaptations to be employed before the point of failure.

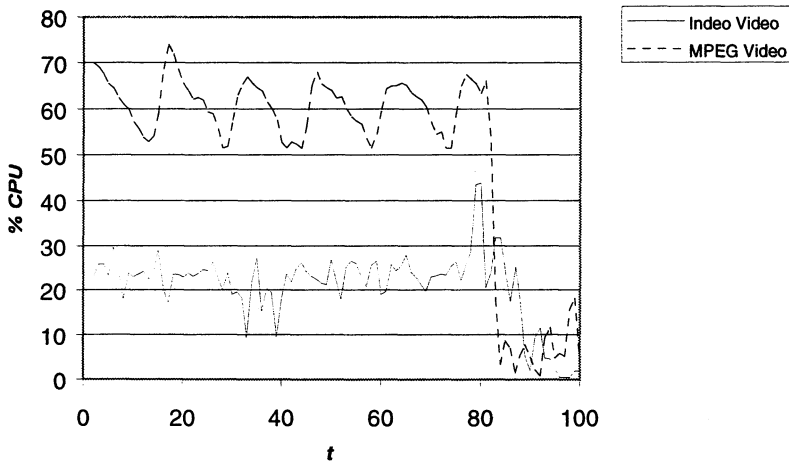


Figure 4 Example CPU Utilisations

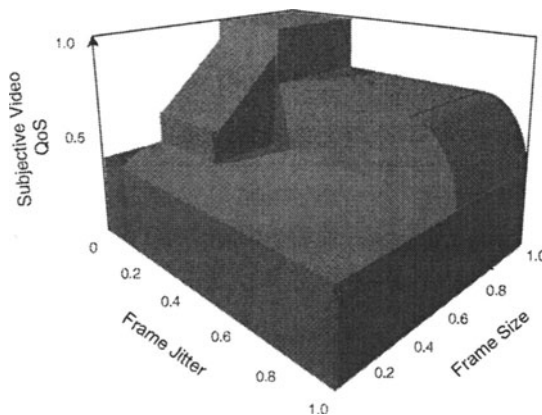
### 3.2 QoS Requirements and Benefit Functions

The objective of any general purpose operating system is to share resources across multiple applications, in a fair and efficient manner. In doing so, the ultimate goal of the system is to fulfil the end-user's requirements, whether that be displaying video without interruption or maintaining a data backup in the event of a system crash. End-user requirements can be defined as *subjective* or *objective*. An example of a subjective end-user requirement is the impairment of transmitted

<sup>2</sup> In this context the term process is used to denote the generation of monitoring information.

video, often used in subjective testing in the engineering community. It is possible to use such perceptual QoS metrics to help optimise multimedia communication services (Verscheure, 1996).

Alternatively requirements can be classified as objective. These requirements are directly associated with a particular QoS metric and hence are more easily described and specified; examples include frame rate and network bandwidth. For this reason, traditional QoS-based resource management systems tend to concentrate on the management of objective requirements. However, there does exist a direct relationship between subjective and objective QoS. Furthermore, this relationship is particularly important in defining, and prioritising, which objective QoS requirements contribute to the overall goal of the system, satisfying the end-user. The relationship between the two can be quantified as a functional expression, known as a *benefit function*, which is a technique originally proposed by (Davis, 1994). The generalised abstraction allows the specification of arbitrary objectives (or subjective QoS requirements), and their relation to resource utilisation and/or objective QoS requirements. In turn, the function can be used by a resource manager to determine which adaptation processes are most beneficial to the end-user. An example of a use of a benefit function is in describing the relationship between frame jitter and frame size and overall subjective quality of service. From the video benefit function shown in figure 5 (Davis, 1994), it is apparent that once finite thresholds of frame jitter and frame rate are reached (0.2 and 0.7 respectively) the benefit of using more resources to increase the resulting subjective QoS is negligible.



**Figure 5** Video Benefit Function

## 1.1 Adaptation Mechanisms

To structure our model, we now define the mechanisms required to support QoS adaptation in a distributed environment. Many resource fluctuations in a distributed

system can be handled implicitly by the processing entities themselves. For example, a video decoder which receives a burst of frames and cannot process these before the next frames are expected may decide to drop a portion of the frames from the burst. In such a case, because the adaptation is very fine grained, the likelihood of the adaptation process being noticeable to the end user is small. However, more sizeable fluctuations may become apparent through resource monitoring and prediction as previously discussed, and given that a system monitoring process has indicated that QoS degradation is occurring, or QoS failure is imminent, ideally the system should adapt its resource utilisation in an attempt to maintain the end-user's level of service. As discussed in section 2, the process of adaptation, particularly in a distributed environment, involves addressing the balance of resources between the clients, servers and the network connections; the role of adaptation processes is to initiate such a balancing.

**Table 2** Adaptation Mechanisms

<i>Mechanism</i>	<i>Technique</i>	<i>Resource Usage Shift</i>
Resource Control	Static Rate Shaping	client/network → null
	Dynamic rate shaping	client/network → server
	Cache Optimisation	server → server
	Priority Adjustment	client_a → client_b
	Scaling	network → server
	Network Service Control	client/server → network network → client/server
End-to-end Reconfiguration	Dual Codec Insertion	network → client/server
	Client Side Coder Insertion	network/server → client
	Server Hand-off	server_a → server_b
	Network 'swap in'	network_a → network_b
	Service provider fault action	network_a → network_a
Explicit Change of Service	Audio channel drop	client/network/server → null
	Video Adjustment	client/network/server → null

We identify three classes of adaptation mechanisms: *resource control*, making fine grained adjustments to individual resources in the distributed system; *reconfiguration*, altering the topology of the end-to-end processing; and *change of service*, allowing the user to prioritise services and adjust as necessary. The majority of adaptation mechanisms fit into one of these classes. Each mechanism has a certain granularity, and each may affect the resource distribution in a slightly different manner. Table 2 offers some examples of adaptation mechanisms and indicates the resulting shift of distribution in resource utilisation. The actual choice

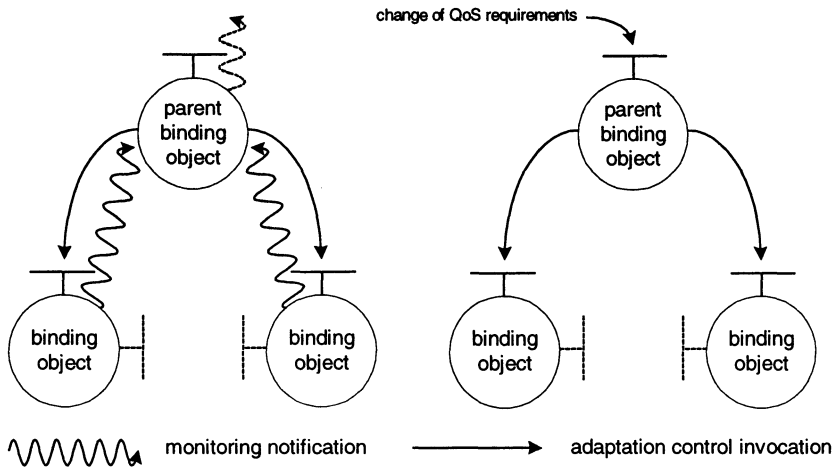
of adaptation mechanism in response to received monitoring indications, is dictated by a set of system-wide *adaptation policies*. An adaptation policy describes, in conjunction with the applications resource capacity regions, what adaptation processes (control, reconfiguration or change of service) should be executed in response to various QoS scenarios. Furthermore, policies can be used in conjunction with the previously discussed benefit functions, allowing the prioritisation of adaptation mechanisms. Policies are applicable from end-to-end and are associated with any of the core distributed resources and their processing configuration. We suggest that the specification of policies should employ open interfacing techniques, aiding extensibility and readily understood programming level abstractions.

## 4 IMPLEMENTATION PERSPECTIVES

This section is concerned with the incorporation of the discussed QoS adaptation techniques into the Distributed Resource Management Architecture (DRMA) currently being developed by Lancaster University and BT Labs (Waddington, 1997). The DRMA platform focuses on the deployment of distributed multimedia applications over multi-service ATM networks and offers a framework for the hierarchical management of end-to-end resources. In addition, the implementation exploits open interfacing and distributed object techniques to aid scalability, flexibility and extensibility.

### 4.1 Adaptive Bindings

Within the DRMA, adaptation, monitoring and control processes are all engineered as a set of hierarchically structured *binding objects* (see figure 3). A binding object is a particular class of component which is used to abstract the functionality of a 'binding' between one or more source and sink components; a binding represents any form of communication (in the network or in the end-system) between these components. The combination of binding objects and other processing components is used to form the distributed application. Because the binding objects are hierarchically distributed, the adaptation and monitoring mechanisms also become distributed, thus avoiding the problem of centralised control and management. Each binding object maintains a set of interaction interfaces which describe the level of service offered through the binding communications, and furthermore define the point-to-point QoS constraints. The role of the binding object is to maintain the desired QoS, as specified by its interfaces, and in doing so it may, if required, employ monitoring and adaptation mechanisms. More detail on distributed binding objects is given in (Waddington, 1997).



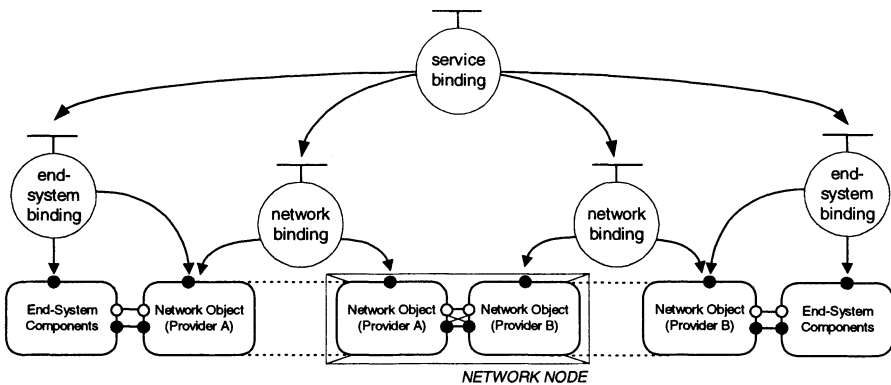
**Figure 6** Explicit Hierarchical Adaptation

At the lowest level of the binding hierarchy, atomic processing components are linked together to form an end-to-end processing chain. Because adaptation and control at this level are very fine grained, the adaptation processes are both implicit (autonomous control adjustments) and explicit (control adjustments from parent bindings). As a consequence, care must be taken to avoid implementing adaptation policies which conflict or which may result in continuous counteractions. Explicit adaptation actions are usually triggered as a result of either degradation notifications or higher level changes in service requirements (see figure 6). In the former scenario, indications of QoS failure or degradation are passed from the child to the parent bindings. Adaptation policies are then parsed to determine what actions should be taken, and then any necessary adaptation processes are executed by the local binding object. If the binding object is unable to correct the degradation through adaptation (by a lack of suitable adaptation policies), or attempts at adaptation have failed, then the QoS notification is forwarded to the parent binding object (all notifications are passed through clearly defined QoS-feedback object interfaces). This process is continued until either successful adaptation actions are executed, or in the event of un-correctable failure the end-user is notified.

### *Example Adaptive Binding*

As previously discussed in section 2.1, coarse grained adaptations occur within the higher level binding objects, such as the service binding. We now describe an example adaptive binding which is used to provision a multimedia streaming service across multi-domain ATM network. The binding hierarchy, as illustrated in figure 7, is composed of a service binding which delegates the end-to-end communications to concatenated end-system and network bindings.

During the initial setup, the service binding is responsible for mapping application level QoS requirements onto end-system and network level requirements (a process often referred to as QoS mapping). It is likely that many such mappings exist for a particular end-to-end service. Nevertheless, in determining the choice of end-to-end resource, the service binding object must make consideration of current resource availability (determined through monitoring) and their cost, a metric which is particularly relevant to network resources. Once a mapping has been determined the binding objects are instantiated and their QoS requirements exchanged. The next phase of the setup involves each individual binding object carrying out further QoS mapping to determine what resources are required to maintain its previously agreed point-to-point requirements. If there are insufficient resources, then the binding object must indicate an admission failure to the parent service binding. In our prototype implementation, the end-system binding uses statically defined mapping information to create a chain of end-system processing components (such as decoders and renderers) which meet the QoS requirements previously requested. In admitting the components, the end-system binding must also ensure that there are approximately (because we are using loose reservation) sufficient available CPU, memory, and other end-system resources. The setup of the network binding objects is slightly different. The network binding must employ some form of connection setup and routing, to establish a point-to-point QoS constrained network service. In our prototype implementation, we have used the ATM forum's UNI/NNI based signalling to create a guaranteed service to a Winsock2/AAL5 protocol stack.



**Figure 7** Example Adaptive Binding

During the lifetime of the service binding, there are potentially changes in the state of end-to-end resources which may in turn require QoS adaptation. For instance, one likely scenario is that the level of network service that was initially reserved is not sufficient for the application's streaming requirements (maybe video frames are being lost). Such mismatches in reserved or available resources, and the actual required resources, are indicated through QoS monitoring fed back from the

network objects (see section 4.1). On receipt of QoS degradation notifications, it becomes the responsibility of the service binding to initiate QoS adaptation. The choice of adaptation is made through a finite state machine representing the originally discussed resource capacity region. This is used to determine which adaptation policies are valid for the given level of service. If the binding is unable to make any suitable adaptation, then the component must indicate QoS failure to the application. Within this context likely adaptation scenarios would be adjustments in the network QoS (maybe through the setting up of a new ATM connection and carrying out a hot swap) or alternatively the incorporation of compression components in the end-systems. In some cases adaptation actions may take the form of reconfiguration. For example, consider the scenario where the initially chosen network service provider can only support a limited 25Mbps connection and a requirement of 26Mbps becomes evident. In this case the service binding object may choose to use an alternative network service provider and release the previously allocated network resources. Finally, in parallel with the previously discussed coarse-grained adaptations, the system is likely to experience various fluctuations in low level QoS, which are counter-acted through intrinsic fine grained adaptation actions.

## 5 CONCLUSION

We have proposed a general model for the support of QoS-based adaptation and resource management in distributed multimedia systems. Our perspective is across the complete end-to-end co-ordination and control of network and end-system resources, supporting end-to-end QoS constrained processing and communications. The general model of QoS adaptation incorporates the following principles:

- QoS provisioning through distributed resource management;
- Hierarchical approach to QoS management, leading to scalability;
- Combination of 'loose' resource scheduling and dynamic adaptation policies;
- Adaptation through both monitoring and prediction;
- Indications of end-user requirements through functional modelling of benefits.

We have also made progress towards the implementation of our proposed QoS adaptation mechanisms, using QoS adaptation techniques in the development of the prototype Distributed Resource Management Architecture (DRMA); DRMA provides a distributed platform for the development and deployment of QoS-constrained continuous media applications. The system, based on Windows NT, uses distributed object programming methods and hierarchical QoS management to support adaptive bindings which transparently encapsulate the proposed resource monitoring and adaptation mechanisms. Our future work is directed towards a fuller implementation of the Distributed Resource Management Architecture together with further support for QoS management and QoS adaptation, in both the network and end-system bindings.



## 6 ACKNOWLEDGEMENTS

We would like to acknowledge the kind support of BT Labs in funding this research under their Management of Multi-service Networks University Research Initiative (BT-URI).

## 7 REFERENCES

- Anderson, D.P. (1993) Metascheduling for Continuous Media, *ACM Transactions on Computer Systems*, Vol. 11, No. 3, pp. 226-252.
- Bolot, J.C. and Turetti, T. (1996) Adaptive Error Control for Packet Video in the Internet, *IEEE Signal Processing Society, Proceedings of the International Conference on Image Processing (ICIP)*, Lausanne.
- Campbell, A. and Coulson, G. and Hutchison, D. (1994) A Quality of Service Architecture, *ACM Computer Communications Review*.
- Chatterjee, S., Sydir, J. and Sabata, B. (1997) Modelling Applications for Adaptive QoS-based Resource Management", *Proceedings of the 2<sup>nd</sup> IEEE High Assurance Systems Engineering Workshop*, Bethesda, Maryland.
- Dang Tran, F. and Perebaskine, V. (1995) TORBoyau: Architecture and Implementation, General Leclerc, Issy-les-Moulineaux, France.
- Davis, M. and Downing, A. (1994) Adaptable System Resource Management for Soft Real-Time Systems, *Symposium on Command and Control Research and Decision Aids*, Monterey, California.
- Degermark, M., Kohler, T., Pink, S. and Schelen, O. (1995) Advance Reservations for Predictive Service, *Proceedings of 5th International Workshop on Network and Operating System Support for Digital Audio and Video*, Durham, New Hampshire.
- Edwards, C., Hutchison, D. and Waddington, D. (1998) Open Interface Support for Heterogeneous Network Services, to be presented at the Third European Conference on Multimedia Applications, Services and Techniques (ECMAST'98), Berlin.
- Huard, J., Inoue, I., Lazar, A. and Yamanaka, H. (1996) Meeting QoS Guarantees by End-to-end QoS Monitoring and Adaptation, *Workshop on Multimedia and Collaborative Environments of the Fifth IEEE International Symposium On High Performance Distributed Computing*, Syracuse, NY.
- Hyman, J., Lazar, A. and Pacifici, G. (1991) Real-time Scheduling with Quality of Service Constraints, *IEEE Journal on Selected Areas in Communications*, Vol. 9, pp. 1052-1063.
- ISO/IEC JTC 1/SC21 (1997) "Quality of Service in ODP – Attachment 1"
- Lazar, A., Ngoh, L. and Sahai, A. (1995) Multimedia networking abstractions with quality of service guarantees, *Proceedings of the SPIE Conference on Multimedia Computing and Networking*, San Jose, CA.

- Nahrstedt, K., Hossain, A. and Kang, S. (1995) Probe-based Algorithm for QoS Specification and Adaptation, University of Illinois, QoS Workshop, Paris.
- Ott, M., Michelitsch, G., Reininger, D. and Welling, G. (1997) An Architecture for Adaptive QoS and its Application to Multimedia Systems Design, to appear in Special Issue of Computer Communications on Building Quality of Service into Distributed Systems, C&C Research Laboratories, NEC, USA.
- Verscheure, O. and Hubaux, J. (1996) Perceptual Video Quality and Activity Metrics: Optimization of Video Service Based on MPEG-2 Encoding, Springer Series LNCS 1185, Proceedings of 3<sup>rd</sup> International COST237 Workshop, Barcelona, Spain.
- Waddington, D., Edwards, C. and Hutchison, D. (1997) Resource Management for Distributed Multimedia Applications, Second European Conference on Multimedia Applications, Services and Techniques (ECMAST '97), Milan.
- Waddington, D. and Coulson, G. (1997) A Distributed Multimedia Component Architecture, Proceedings of the 1<sup>st</sup> International Workshop on Enterprise Distributed Object Computing, Gold Coast, Australia.
- Wolf, L., Delgrossi, W. L., Steinmetz, R., Schaller, S. and Wittig, H. (1995) Issues of Reserving Resources in Advance, Proc. 5th Intl. Workshop on Network and Operating System Support for Digital Audio and Video, Durham, New Hampshire.

## 8 BIOGRAPHY

Daniel Waddington is a final year Ph.D. student, with the Distributed Multimedia Research Group. He is currently working as a Research Assistant on the British Telecom University Research Initiative (BT-URI) project. On the BT-URI, Daniel is looking at issues of end-to-end QoS management and a framework for service provision. His primary Ph.D. research interests are Distributed Object Computing and its support for distributed multimedia services.

David Hutchison is Professor of Computing at Lancaster University and has worked in the areas of computer communications and distributed systems for the past 15 years. He has completed many UK and European funded research contracts and published over 100 papers as well as writing and editing books on these areas. He has just finished a years sabbatical leave as a visiting academic at HP Labs in Bristol, UK, at EPFL in Lausanne, Switzerland, and at BT Labs in Ipswich, UK.

# **A Dynamic Sender-Initiated Reservation Protocol for the Internet**

*P. P. White, J. Crowcroft*

*Department of Computer Science*

*University College London*

*Gower Street*

*London WC1E 6BT*

*England*

*phone: +44 171 419 3701, +44 171 380 7296*

*fax: +44 171 387 1397*

*email: p.white@cs.ucl.ac.uk, j.crowcroft@cs.ucl.ac.uk*

## **Abstract**

In this paper we discuss the need for resource reservation in the Internet and examine some of the strengths and weaknesses of RSVP, which is currently the most popular of Internet reservation protocols that have been developed. The deficiencies of RSVP motivate our design of a new resource reservation protocol which uses dynamic sender-initiated reservations to achieve a highly bandwidth-efficient reservation mechanism with excellent scalability with regards to round trip time, data rate and number of hosts.

## **Keywords**

**Resource Reservation, Quality of Service**

## 1 INTRODUCTION

It is clear that the current Internet which was founded upon the concept of 'best-effort' datagram delivery must be enhanced in some way in order to accommodate the changing communications environment. In particular there is a growing demand for real-time applications which have specific Quality of Service(QoS) requirements, especially with regard to end-to-end delay and minimum bandwidth, both of which cannot be guaranteed in the current Internet using traditional connectionless best-effort delivery. Furthermore as the World-Wide-Web is increasingly used for business there is a growing number of users for whom delay bounded access of information is important.

In response to the changing requirements of Internet users, much attention has focussed on the use of resource reservation as a means of providing selected data flows with special QoS commitments in accordance with their needs. Under such a framework it is likely that special QoS delivery would be the exception rather than the rule with the majority of Internet traffic continuing to receive the 'default' best-effort mode of delivery. The special QoS required by a specific data flow can be realised by reserving resources(bandwidth, buffer space) and installing appropriate scheduling behaviour in each router along the end-to-end path followed by the data flow. Such mechanisms require admission control at the individual intermediate nodes to ensure that the request for reservation is only accepted and installed provided sufficient resources are available. In addition, per-flow state<sup>1</sup> in the intermediate nodes will usually be required in order to identify the flows to receive special QoS as well as the QoS to be received.

In order to allow users to invoke special QoS delivery on demand for a data flow several protocols have been developed to enable users to communicate their QoS needs to the intermediate routers along the data path in an IP internetwork. The majority of these protocols initiate the set up of flow-specific reservation state in intermediate routers, a notable exception being the approach described in (Almesberger, 1997) whereby no per-flow reservation state is set up in routers which instead record their reservation commitments as a whole per output port. While this approach potentially offers very good scalability characteristics for a large number of flows, it is dependent upon a certain degree of trust among end hosts not to exceed their indicated traffic levels unless per-flow policing is applied at the network access point. In addition, the approach is only able to offer end applications an approximate minimum bandwidth without any quantitative guarantees on loss or delay and so may not be suitable for applications with stringent QoS requirements such as Distributed Interactive Simulation (Seidensticker, 1997).

Of the reservation protocols that set up flow-specific reservation state, an early example in the Internet is the Stream Protocol, ST (Forgie, 1979) which was

---

<sup>1</sup> The introduction of per-flow state is a significant departure from the initial Internet design philosophy of a pure connectionless network with no per-flow state in the intermediate routers.

limited to unicast reservations. Although its successors, ST-II (Topolcic, 1990) and the more recent ST2+ (Delgrossi, 1995) can handle both multicast and unicast reservations as well as possessing many improvements over ST, the ST group of protocols has attracted little commercial interest. By contrast, another reservation protocol, RSVP (Braden, 1996) has received significant industry support and with good reason. Unlike the ST protocols, RSVP reservation state is soft-state and will time-out in the absence of any refresh reservation requests within a certain time period. This so-called soft-state nature of RSVP provides a very simple failure recovery mechanism over a wide range of fault scenarios and helps to retain much of the robustness that has helped to make IP so successful. The soft-state approach where the end applications are responsible for maintaining the flow-specific router state leads to a significant reduction in complexity compared to a hard-state approach where the network is responsible for maintaining the flow-specific router-state. RSVP has other notable architectural differences compared to the ST protocols such as receiver-initiated rather than sender-initiated reservations<sup>2</sup>. The initial design of RSVP was to a large extent influenced by the needs of multicast conferencing applications although its intended use is now much broader.

While RSVP is concerned merely with signalling the end application's reservation requests to the intermediate nodes, it is the special QoS delivery models<sup>3</sup> that define the node behaviour required to meet the signalled special QoS objectives. The Integrated Services Working Group(intserv) of the IETF(intserv 1998) has standardised several special QoS delivery models while the Integrated Services over Specific Lower Layers(issl) Working Group of the IETF(issl 1998) has developed ways of mapping this network layer QoS onto specific link layer technologies such as ATM, IEEE 802 and Ethernet.

In parallel with the recent Internet growth much interest has been generated by Asynchronous Transfer Mode(ATM), a technology designed from the outset with end-to-end QoS in mind. A necessary component in ATM networks for achieving QoS on demand is a signalling protocol in order to request resource reservations in the intermediate nodes of the end-to-end path. More traditional ATM signalling protocols such as ITU's Q.2931 standard for public networks (ITU-T, 1995) or ATM Forum's UNI standards for private networks (ATM Forum, 1996) use end-to-end handshaking to set up an end-to-end reservation before data transfer can take place. A more dynamic and flexible approach is that provided by the ATM Block Transfer/Immediate Transfer(ABT/IT) (ITU-T, 1996) signalling protocol which sends reservations in-line with data and as such is more conducive to efficient bandwidth utilisation than the more static end-to-end handshaking approach.

---

<sup>2</sup> ST-II+ permits both sender and receiver-initiated reservations, ST-II and ST permit sender-initiated reservations only.

<sup>3</sup> At present two delivery models have been standardised, both of which offer applications an end-to-end minimum bandwidth albeit with different assurances. First, Guaranteed Service which offers applications a loss-free service with an end-to-end delay bound. Second, Controlled-Load Service which does not provide any quantitative guarantees on delay or loss, although qualitatively these parameters can be expected to be the same as for best-effort delivery under low network load.

In the next few sections we present a new QoS signalling protocol known as Dynamic Reservation Protocol(DRP) which could be used to set up the IETF's integrated services models 'on-the-fly' in IP internetworks. We outline the benefits of DRP compared to RSVP before presenting details of packet formats and processing rules and our conclusions.

## 2 DYNAMIC RESERVATION PROTOCOL (DRP) OVERVIEW

Our protocol, known as Dynamic Reservation Protocol(DRP) incorporates many principles of RSVP along with the dynamic sender-initiated reservation concept of ABT/IT to achieve the following goals:

- High control dynamics to achieve efficient bandwidth usage for both sender-specific and shared reservations.
- Scalability of router-state with regard to number of senders and receivers.
- Scalable and simple approach to One Pass With Advertising (OPWA)<sup>4</sup>.
- Minimal receiver complexity.
- Minimal number of messages to implement session-wide reservation changes in large-scale multicast sessions.
- Heterogeneity of reservation QoS classes among receivers of the same session.

DRP allows reservations to be set up 'on-the-fly' by sending Reservation packets, RES in-line with the data flow. In this respect, DRP is similar to ABT/IT although, unlike ABT/IT, DRP does not need to make an end-to-end connection before sending its first in-line reservation packet. Also, unlike ABT/IT, DRP does not support the concept of a sustainable cell rate for the data transfer and consequently the probability of acceptance of a reservation request is determined purely by the available resources at that moment in time. As with RSVP, all flow-specific router state that is set up using DRP is soft-state as we believe that this is a key strength of RSVP that can also be used to good effect in DRP.

The scheme also uses Return (RTN) packets that are reverse-routed up the tree to provide the intermediate routers/switches and sender with certain feedback and end-to-end path information. DRP is applicable to both unicast and multicast scenarios but in the following sections we concentrate on the more complicated multicast case.

## 3 DRP DESIGN PRINCIPLES

### 3.1 Sender initiated reservations

The use of in-line reservation packets allows the sender to set up new reservations, or alter existing reservations, on demand at any point in the data transfer. This

---

<sup>4</sup> This is a term introduced by RSVP, to describe a mode whereby all necessary information is made available(advertised) in advance of making a reservation request so that the correct level of reservations necessary to achieve the target end-to-end QoS can be determined and installed in 'one pass' of the reservation message.

makes it possible to achieve a very close match between the instantaneous service provided by the network and the instantaneous requirements of the data flow. As a result, network resource usage can be minimised. These benefits are particularly prominent for stop/start data flows since the resources can be freed during the quiet periods and re-installed on a just-in-time basis at the start of each activity burst. Such action is precluded with both RSVP and traditional ATM signalling<sup>5</sup> approaches, both of which incur a time lag in excess of the round-trip time when modifying end-to-end QoS to reflect a change in the sender's traffic stream characteristics.

Another advantage of using sender-based reservations rather than receiver-based reservations is a reduction in the volume of processing required at intermediate nodes of a multicast tree each time a sender changes its traffic stream characteristics. With RSVP, each time this occurs and the receivers consequently modify their reservations, it is possible for a node to install a reservation due to a request from a particular receiver, only for it to increase the reservation a moment later when a larger request from a different receiver arrives at the same interface. In fact when the multicast tree serves a large number of receivers it is possible that some reservations may be updated several times before settling down to their steady state values. This effect will be particularly prominent for a Guaranteed Service session since each receiver will probably need to request a different reservation bandwidth even if they require the same end-to-end delay bound. By contrast with DRP, a single pass of the RES packet down the multicast tree will typically<sup>6</sup> achieve the new steady state reservations in the on-tree nodes.

### 3.2 Heterogeneity of QoS reservation classes between receivers of the same session

DRP allows the sender to request, 'on-the fly', intserv's Controlled-Load Service (Wroclawski, 1997) and Guaranteed Service (Schenker, 1997) by sending a RES packet in-line with data. The sender designates a 'Ceiling' Reservation class (or Type), CRTs to each data flow block<sup>7</sup> as well as an associated end-to-end QoS level. In addition each receiver specifies a 'ceiling' reservation class, CRT<sub>r</sub> which represents the highest quality reservation class it is willing to receive. The Guaranteed Service reservation class is taken to be the highest quality reservation class, followed by Controlled-Load Service with 'best-effort' (no reservation) being the lowest. Assuming that sufficient end-to-end resources exist, the effective end-to-end reservation class received by a receiver will then be given by MIN(CRTs, CRT<sub>r</sub>). Each receiver is free to change its value of CRT<sub>r</sub> at any time by sending a

---

<sup>5</sup> Traditional ATM signalling (e.g., Q.2931 and UNI) requires end-to-end handshaking.

<sup>6</sup> In the case of Guaranteed Service, DRP may sometimes use feedback to alter certain reservations after the first pass in an attempt to achieve a target end to end delay bound that was not satisfied on the first pass as described in section 3.6.

<sup>7</sup> A data flow defined by the combination of (sender IP address, sender port, destination IP address, destination port, transport layer protocol) can be considered as a series of data flow blocks, each of which may have its own specific QoS requirements.

RTN packet upstream containing the new value of CRT<sub>r</sub>. In addition, the reservation class installed at each on-tree outgoing interface will be the lowest quality reservation class that is necessary to guarantee each receiver their effective end-to-end reservation class as determined by the above rules. This is exemplified in Figure 1.

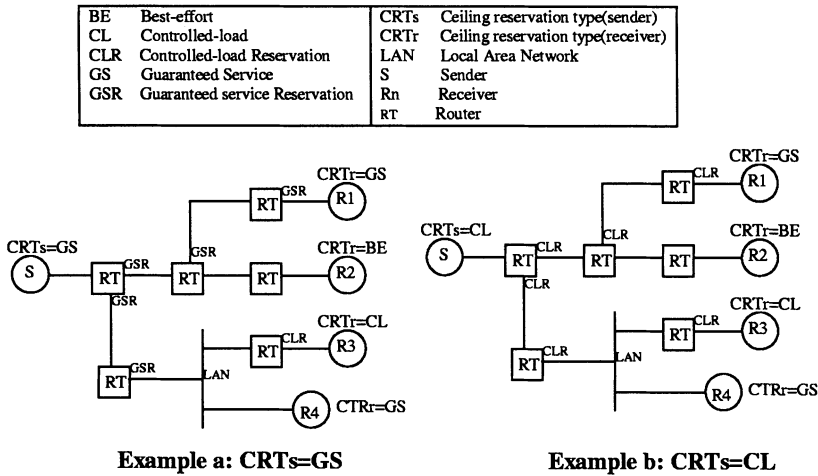


Figure 1: Heterogeneity of reservation classes between receivers of same session.

Table 1 compares the DRP approach to providing end-to-end delay bounds with those of RSVP and ABT/IT. DRP is similar to ABT/IT in the sense that for a given sender to a multicast session the target end-to-end delay bounds will be identical for each receiver<sup>8</sup>. The difference is that with DRP the sender can control what this delay bound will be whereas with ABT/IT the delay bound is a feature of the QoS class provided by the network and cannot be controlled by end nodes. Like DRP, RSVP facilitates end node control of end-to-end delay albeit by receivers rather than senders. RSVP allows receivers finely grained control within a reservation class at the expense of added receiver complexity together with lack of support for reservation class heterogeneity among receivers of a session. By contrast, DRP supports such reservation class heterogeneity in that the sender suggests a reservation class and QoS level for all receivers who then have the option of downgrading QoS class. Although DRP does not offer receivers any control over QoS level within a class, we do not believe such a feature is necessary anyway and certainly not with regard to end-to-end delay bound. We argue that any such end-to-end delay bound is determined by the nature of the sender's traffic stream and as

<sup>8</sup> In the case of DRP we are only referring to those receivers that have actually requested an end-to-end delay bound, i.e. those with CRT<sub>r</sub>=GS.



such the sending application is the node most qualified to specify what it should be. The receivers simply need to be told what this end-to-end delay bound is so that they can set their playout buffers accordingly. Furthermore, removing receiver-control of end-to-end delay in DRP enables merging of RTN messages and RTN state in routers to ensure scalability to large multicast sessions as described in section 3.4.

	RSVP	ABT/IT	DRP
Sender control of delay bound	No	No	Yes
Receiver control of delay bound	Yes	No	No

Table 1: Comparison of different schemes with regard to provision of end-to-end delay bounds

### 3.3 Reservation request admission control.

Apart from the initiator of QoS requests(sender vs receiver) there are two other notable differences between the reservation mechanisms used by RSVP and DRP.

#### 1. Explicit vs implicit reservation requests.

With RSVP, for both Controlled-Load and Guaranteed Service reservations, the request explicitly informs the node of the level of resources to reserve. The same can also be said of a Controlled-Load Service reservation request using DRP. However with a DRP Guaranteed Service request the RES packet requests the level of resources to reserve implicitly by informing the router of the accumulated delay bound thus far, together with the target delay bound and the sender traffic characteristics. Using this information along with path information obtained from RTN packets each router is able to estimate the local reservation required and update the accumulated delay bound in the RES packet accordingly. Each router calculates a local reservation bandwidth which, if also reserved in each subsequent router, will lead to an overall delay bound equal to the target delay bound. However, should any router have insufficient resources to install the calculated local reservation bandwidth then it reserves the most that it can and the attempt is only referred to as a 'reservation failure' if the resultant accumulated delay thus far exceeds the target delay bound. If the attempt is not a so-called 'reservation failure' then the RES message is treated the same regardless of whether the level of local reservation initially calculated could be reserved or it couldn't. This is because even in the latter case the target end-to-end delay bound may still be met since each subsequent router will automatically attempt to reserve more in order to compensate. The action taken in the event of a so-called 'reservation-failure' is discussed next.

#### 2. Action in event of reservation failure.

With RSVP, any request that fails admission control at a router is not propagated any further along its path towards the sender(s) and a ResvErr message is sent to

affected receiver(s). By contrast, whenever a DRP node cannot satisfy the calculated local reservation, and the maximum level of resources that it can reserve is so low that it prevents the target end-to-end QoS from being satisfied, the request is not rejected. Instead, the node reserves as much resources as possible and sets specific QoS violation bits in the RES header while updating the other header fields in the usual manner before propagating the RES message down the distribution tree.

In the event of a DRP Controlled-Load Service 'reservation-failure', the node sets a bit, known as the QoSvoid bit, to 1. The RES packet is handled in the usual way by all subsequent routers encountered, although the presence of the non-zero QoSvoid bit will be an indication to receivers that the end-to-end QoS could not be achieved.

In the event of a Guaranteed Service request where the node could not reserve more than the mean rate of the sender's traffic, it becomes impossible to guarantee either lossless transmission or conformance to the target delay bound, or even the Controlled-Load Service. Consequently three flags should be set in the RES packet, namely the delayvoid, lossvoid and QoSvoid bits. When downstream routers see a RES packet with (CRTs=GS, delayvoid=lossvoid=1) then they take the 'effective CRTs' to be Controlled-Load Service(CL) and attempt to install a Controlled Load Service Reservation.

In the event of a Guaranteed Service request where lossless transmission has not yet been precluded<sup>9</sup> but the accumulated bound at a node exceeds the target delay bound for the first time, the action taken is as described in section 3.6.

In the case of Guaranteed Service reservations in a large multicast tree, there are some interesting differences between DRP's sender-based reservations and RSVP's receiver based reservations. To illustrate these differences we refer to the example topology of Figure 2. This shows the logical connectivity of a multicast session between a sender, S and two receivers, R1 and R2. These end nodes are interconnected via routers, r1-r3 and all links are 10Mbps Ethernet. The exported C and D error terms (Schenker, 1997) from the routers are shown together with the token bucket parameters of the Sender Tspec. We will assume that both receivers, R1 and R2 require a queuing delay bound of 300ms to sender S. With RSVP, each receiver calculates an Rspec that to be reserved in each router along the end-to-end path in order to achieve its delay bound. In this example, R1 calculates an Rspec of 325.3Kbytes/s while R2 calculates an Rspec of 490.89 Kbytes/s. At router r2 these two requests are merged so that the Rspec propagated to router r1 is 490.89Kbytes/s. Packets from S to receiver R1 will now experience a reservation bandwidth of 490.89kbyte/s in router r1, interface 2 rather than the requested 325.3 Kbytes/s. This will cause a reduction in R1's end-to-end delay meaning that theoretically the bandwidth reserved for R1 in r2, interface 2 could be decreased from the initially calculated value of 325.3 Kbytes/s while still achieving R1's end-to-end delay bound. However R1 does not facilitate such a mechanism and in this example R1's end-to-end delay bound will be less than, rather than equal to, that requested. By contrast, DRP keeps a running total of end-to-end delay bound

---

<sup>9</sup> That is, every node so far has been able to reserve in excess of the mean rate of the sender's traffic

which it updates at each hop and uses to calculate the local reservation required to stay on course for the desired end-to-end delay bound. As a result, in this example DRP automatically readjusts the reservation level in r2, interface 2 where 247.7 Kbytes/s is then reserved rather than 325.3 Kbyte/s as in RSVP.

For multicast examples such as this where the routers and links are homogeneous(same values of C, D error terms and link propagation delay) RSVP will never use fewer resources than DRP. However in environments with heterogeneous routers and links the matter is not as straightforward. With DRP, in a multicast environment the bandwidth to be reserved at each node is calculated based on, among other things, worst-case merged(see section 5.3)path characteristics received from RTN messages. The effect of this worst-case merging can be for DRP to make an over-estimation in the local reservation. Any such over-estimation will cause a reduction in the local node queuing delay. In turn this will mean that DRP allows an increase in the local queuing delay at nodes further downstream whose reservations will then not be as high. This ‘skewing’ of the bandwidth reservation pattern in multicast sessions whereby nodes closer to the sender are more likely to over-estimate their local reservations can theoretically cause an increase in the overall reservation bandwidth required in the multicast tree. This is an area for further study.

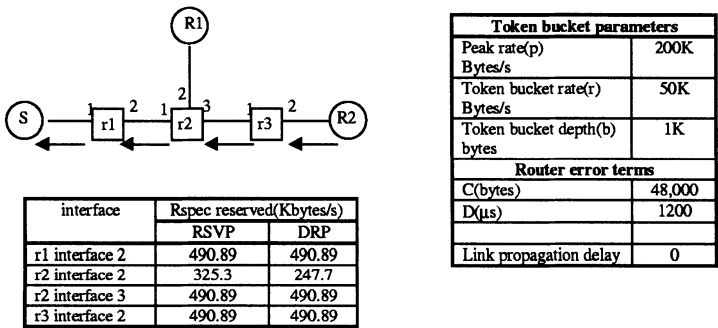


Figure 2: Guaranteed Service Reservations using RSVP and DRP

3.4 Merging of RTN messages

DRP uses RTN messages which are reverse-routed up the distribution tree from receiver(s) to sender(s) for the following purposes:

- 1. To accumulate certain path characteristics information which is used by a node when calculating the level of resources to reserve.
- 2. To allow a receiver to downgrade its received reservation class below that suggested by the sender.
- 3. Optional feedback information that may be used to convey information to intermediate routers in cases where the end-to-end delay bound was not satisfied in the first pass of a RES message.

With respect to 1., RTN messages fulfil a similar role to Path messages in RSVP.

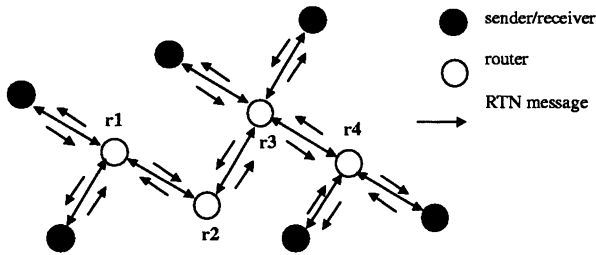


Figure 3: DRP RTN messages on shared tree per refresh period in the steady state.

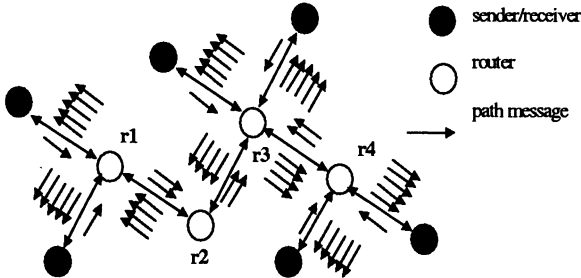


Figure 4: RSVP Path messages on shared tree per refresh period in the steady state.

For large-scale multipoint-to-multipoint applications the use of a single shared tree, such as a Core Based Tree (CBT), for all senders to a multicast group will consume far less resources (Billhartz, 1997) than a separate source-based tree for each sender to the group. Because of this, a single shared tree is likely to be preferable to a mesh of source-based trees in such scenarios. In such cases DRP displays much more favourable scalability characteristics than RSVP. With DRP, full merging of RTN messages is possible and ensures that the number of RTN messages on each link of a shared tree in the steady state is never more than two (one in each direction) every refresh interval as shown in Figure 3. By contrast, with RSVP the total number of Path messages on each link of a shared tree per refresh period in the steady state is equal to the number of senders as shown in Figure 4. However perhaps a more important benefit of the DRP approach is the fact that the number of RTN state entries in each on-tree router is equal to the number of on-tree logical interfaces and so never becomes an issue no matter how many hosts are sending to the group. By contrast, with RSVP the number of Path state entries in each router and end-host of a multicast shared-tree is equal to the number of senders to the group and consequently may become excessive for large-scale multipoint-multipoint applications.

### 3.5 Shared-Session Reservations and Intra-Session Reservation Style Heterogeneity

RSVP supports shared-style reservations which match on multiple senders and are usually used on the understanding that only one sender will be active at once. Although this will yield resource savings compared to a number of sender-specific reservations, shared-style reservations that are set up using RSVP can still be sub-optimal for 2 reasons. First, the reservation stays in place during quiet periods. Second, during active periods the reservation may sometimes or always be larger than necessary to meet the agreed QoS for the data flow currently using it (White, 1998). Both of these inefficiencies are obviated with DRP if the shared-session<sup>10</sup> is handled using sender-specific reservations that are installed and torn down on-the-fly at the start and end of each activity burst. However, in such cases it may be possible for end users to detect a degradation in QoS at the start of each activity burst due to the finite time required to install the 'just-in-time' sender-specific reservation. One way in which such QoS disruption could be minimised is to use what we refer to as a 'simple shared reservation' which would apply to all senders to the session and would be left in place during quiet periods but modified at the start of an activity burst each time the sender to the session changed. With this approach QoS disruption would only occur at the start of an activity burst for a new sender whose reservation requirement was greater than that of the previous sender, but even in this case the QoS disruption would be minimised because of the presence of the now free reservation from the previous sender that can be used as a starting point for the 'just-in-time' reservation request from the new sender to build upon.

While the 'simple shared reservation' mechanism just described would work well in the true 'shared-session' case where there is never more than a single active sender at any one time, it would suffer from under or over-reservations<sup>11</sup> in cases where it is possible for multiple senders to be simultaneously active which might occur in the absence of appropriate conference control mechanisms. This deficiency of a simple shared reservation approach is highlighted in Figure 6 for the example traffic pattern of Figure 5. Bearing these potential hazards in mind, DRP provides an alternative reservation mode to the standard sender-specific(SS) mode known as Sender-Specific with Residue(SSR). In SSR mode each sender makes a reservation at the start of each activity burst and sends a teardown request at the end of the activity burst. When a sender's teardown request<sup>12</sup> reaches an outgoing interface of a router the SSR reservation of the sender will only be removed if at least one other SSR reservation for the session is in place in the

---

<sup>10</sup> where only one sender to the session transmits at once.

<sup>11</sup> That is, over-reservations in addition to the 'over-reservations' present when the reservation stays in place during quiet periods.

<sup>12</sup> A teardown request is simply a RES packet with the reservation level set to 0. It indicates to the intermediate routers that the sender no longer requires a reservation for its data packets.

router at that outgoing interface. Otherwise the sender's SSR reservation is left in place, but a status flag associated with the reservation is set from 'active' to 'passive' state.

Figure 6 illustrates the operation of SSR mode for the traffic pattern of Figure 5. When only one sender is active at once, the operation of SSR mode is essentially the same as with a 'simple shared reservation' and so will suffer from resource wastage when all of the senders go simultaneously quiet<sup>13</sup>. However should the senders go simultaneously quiet for extended periods of time the soft-state nature of the reservation will cause it to eventually timeout and be removed. In cases where more than one sender is simultaneously active, the operation of SSR mode is essentially the same as SS mode and so cannot suffer from the under-reservation problem that exists with the 'simple shared reservation'.

A notable advantage of DRP compared to RSVP is that DRP allows co-existence of both modes of reservations within the same multicast session while with RSVP each receiver within a given multicast session must choose the same reservation style. The way in which co-existence of reservation modes within the same multicast session is accommodated in DRP is summarised as follows.

**When a reservation request arrives at an on-tree incoming router interface** it is copied to each on-tree outgoing interface where the following steps are applied:

*If reservation for that sender already exists*

- Set reservations's mode flag(0=SS, 1=SSR) according to mode field in RES packet.
- Adjust reservation level to value indicated in RES packet.
- Set reservation's status flag to 1 (active).

*Else if mode field in RES packet indicates SS*

- Create a new reservation according to filter spec and value indicated in RES packet.
- Set reservation's mode flag to indicate SS.
- Set reservation's status flag to 1 (active).

*Else if a SSR reservation exists with state = 'passive'*

- Set filter spec of that reservation to the sender of the RES packet.
- Adjust reservation level to value indicated in RES packet..
- Set reservation's status flag to 1 (active).

*Else*

- Create a new reservation according to filter spec and value indicated in RES packet.
- Set reservation's mode flag to indicate SSR.
- Set reservation's status flag to 1 (active).

---

<sup>13</sup> For example in a multimedia conference if the audio channel used a different multicast group to the other multimedia traffic components there might be significant periods of time where the audio channel was quiet. By contrast in an audio-only conference the channel is unlikely to be quiet for any lengthy period of time.

**When a reservation teardown arrives at an on-tree incoming router interface** it is copied to each on-tree outgoing interface where the following steps are applied:

If reservation mode is SS

- Remove reservation

Else if total number of installed SSR reservations including this one is greater than one

- Remove reservation.

Else

- set reservation's status flag to 0(passive)

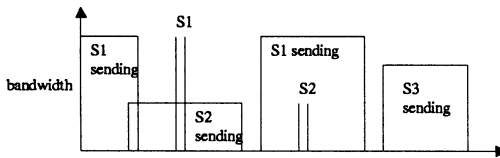


Figure 5: Traffic Pattern for a shared session with some sender transmission overlap

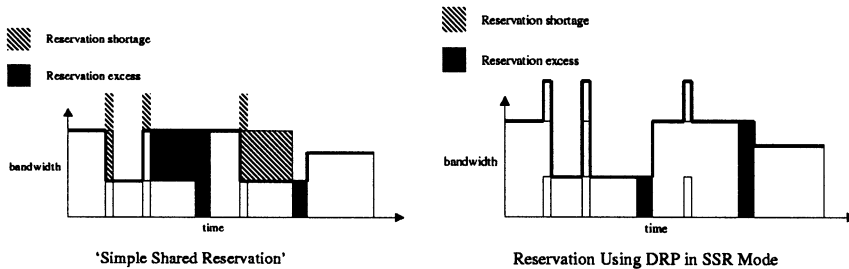


Figure 6: Bandwidth reserved for a shared session with some sender transmission overlap.

### 3.6 Using feedback to increase the probability of achieving end-to-end delay bound

In the case of Guaranteed Service reservations the adoption of a strategy whereby an end-to-end reservation is only permissible by installing an equal reservation in each router reduces the chances of meeting a target end-to-end delay bound. This characteristic has been noted by the designers of Guaranteed Service and exploited to a reasonable degree through the introduction of a slack term (White, 1997) into the reservation flow specification. Use of the slack term enables higher reservations to be made between the receiver and the bottleneck router to compensate for the increase in delay incurred by the lower reservation in all routers between, and including, the bottleneck router and the sender. However it does not permit the reservation to be increased once it has passed through the bottleneck

router on its way towards the sender. Such a restriction can sometimes prevent RSVP from achieving the target delay bound even on a path that actually contains enough resources to meet the target end-to-end delay bound.

Unlike RSVP, DRP employs cooperation and feedback between routers to ensure that if a given end-to-end path is capable of supporting a specific target delay bound then DRP will always meet the target delay bound. In the example of Figure 7 each router is able to reserve a bandwidth in excess of the mean rate of the sender's traffic but at router r3, accumulated delay for the first time is in excess of the target delay bound,  $D_t$ . Consequently, router r3 sets a bottleneck flag associated with its local reservation as well as setting bottleneck and delayvoid flags in the forwarded RES message. When r4 receives the RES message it notices that the delayvoid flag has been set to 1 and so as a result reserves the maximum reservation that it can (subject to any installed policy decisions) in order to minimise its contribution to the accumulated delay. When R receives the RES message it notices that the target delay bound has been exceeded and immediately issues a RTN packet containing the amount by which the target delay has been exceeded in a field in the packet known as the excess delay field. In addition, a bit in the packet known as the bottleneck bit, is set to 0. This RTN packet is reverse-routed up the tree but is ignored at each router until its bottleneck flag has been set to 1 which will occur when it reaches the interface of a router, r3 in this example, in which the bottleneck flag has been set to 1 for the installed reservation. The RTN packet will then travel hop by hop towards S with an attempt being made at each hop to eliminate the excess delay or at least reduce it as much as possible by increasing the level of the local reservation on the appropriate outgoing interface. If a router succeeds in reducing the excess delay to zero then the RTN packet will cause no further alterations in local reservations on the rest of its journey towards S. In this example, r2 is able to increase its local reservation and cause a reduction in its local queueing delay of  $d_1$  which is then subtracted from the excess delay field before sending the RTN packet to r1 which manages to increase its local reservation sufficiently to completely eliminate the excess delay. The target end-to-end delay bound has now been achieved.

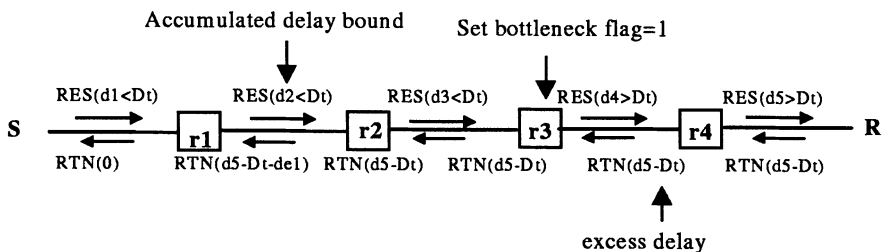


Figure 7: Use of DRP feedback mechanism to maximise chances of meeting target delay bound for Guaranteed Service.

In the next two sections we present details of the main fields required in RES and RTN packets together with the processing rules in order to provide the basic functionality of DRP described in the previous sections.



## 4 RESERVATION(RES) MESSAGE

The IP destination address of the IP datagram encapsulating a RES message is equal to the session destination address while the IP source address is equal to the initial sender of the RES packet. The IP router alert option is used to ensure that intermediate nodes intercept and process the RES packets.

### 4.1 RES message Common Part

- **Session** – (object defined in the RSVP protocol) – it contains the destination address, transport layer protocol identifier and transport layer destination port.
- **Phop** - (object defined in the RSVP protocol) – it is the identity of the last DRP-capable logical outgoing interface to forward this message. The Phop object consists of the pair (IP address, logical interface handle) and is required to install Phop state in the router to ensure correct reverse routing of RTN messages.
- **Sender Template** - (object defined in the RSVP protocol) – it is a filter specification identifying the sender. It contains the IP address of the sender and optionally the sender port(in the case of Ipv6 a flow label may be used in place of the sender port)
- **timestamp** field - this is stamped with the time of the local node clock just before being forwarded to next hop(s) down the distribution tree. It is used to calculate dnext as described in (White, 1998).
- **CRTs field(2 bits)** - this identifies the ceiling reservation class of the sender. 11 indicates Guaranteed Service, 10 indicates Controlled-Load Service, and 00 indicates best-effort. 01 is currently unspecified although may at some time be used for a new service with quality in between best-effort and Controlled-Load Service.
- **Tspec** describing sender's traffic characteristics using the following token bucket representation as described in (Schenker, 1997)
  - p = peak rate of flow (bytes/second)
  - b = bucket depth (bytes)
  - r = token bucket rate (bytes/second)
  - m = minimum policed unit (bytes)
  - M = maximum datagram size (bytes)
- **end2end delay** field - this gives the current delay from when a packet was transmitted by the initial sender until it is due to arrive at the incoming interface of the current next hop.
- **Mode** field(1 bit) – this identifies the reservation mode. A value of 0 indicates SS mode while a value of 1 indicates SSR mode.
- **QoSvoid** bit – if set to 1 this indicates that no QoS guarantees can be offered.

## 4.2 RES message Guaranteed Service object

If CRTs = 11(Guaranteed Service) the RES packet will also contain a Guaranteed Service object comprising the following:

- **CSum** - accumulation of C values since last upstream reshaping point (see (Schenker, 1997)).
- **DSum** - accumulation of D values since last upstream reshaping point (see (Schenker, 1997)).
- **target-bound** field which indicates the target end-to-end delay of the sending application.
- **accumulated-bound** field which indicates the installed delay bound between sender and the incoming interface of the current next hop.
- **Flags** field containing
  - **delayvoid bit** (If set, this bit is an indication to the receiver that the target delay bound cannot be guaranteed)
  - **lossvoid bit** (If set, this bit is an indication to the receiver that a loss-free service cannot be guaranteed)

## 4.3 Node Processing of RES messages

When a node receives a RES packet for an end-to-end reservation attempt at which QoS violation has already occurred or which occurs following processing of the RES packet, the behaviour of the node is as described in sections 3.3 and 3.6. Otherwise the processing of the packet is as described in the remainder of this section.

Upon receipt of the RES message the node passes it to admission control which then determines the reservation that needs to be made at each of the outgoing interfaces. The reservation class is given by MIN(CRTs, CRT<sub>r</sub>) where CRT<sub>r</sub> is obtained from the Merged RTN State Entry(MRTNSE) for the appropriate outgoing logical interface as described in the next section.

If the reservation request is for the Controlled-Load Service then the reservation is governed entirely by the sender Tspec contained within the RES message. By contrast if the reservation request is for Guaranteed Service then the reservation is described by the combination of the sender Tspec and a reservation bandwidth, R that the admission control mechanism needs to determine using the following equations as given in (Schenker, 1997).

$$Q_{delay_{end2end}} = \frac{(b-M)(p-R)}{R(p-r)} + \frac{(M+C_{tot})}{R} + D_{tot} \quad (\text{case } p > R \geq r). \quad (1)$$

$$Q_{delay_{end2end}} = \frac{(M+C_{tot})}{R} + D_{tot} \quad (\text{case } R \geq p \geq r). \quad (2)$$

Admission control obtains the parameters  $M$ ,  $p$ ,  $b$  and  $r$  from the sender  $T_{spec}$  contained in the RES message. The value of  $C_{tot}$  is given by the sum of the router's local  $C$  value and the merged  $C_{tot}$  value as obtained from the MRTNSE (see next section) for the relevant outgoing interface. Likewise the value of  $D_{tot}$  in the above equations is given by the sum of the router's local  $D$  value and the  $D_{tot}$  value in the MRTNSE for the relevant outgoing interface. To obtain the value of  $Q_{delay}$  to insert into the above equations, admission control uses the relationship given in equation (3).

$$Q_{delay} = \text{target-bound} - \text{accumulated-bound} - d_{next} - \text{propdelay}. \quad (3)$$

where the target-bound and accumulated-bound are obtained from the corresponding fields in the RES packet, and propdelay and  $d_{next}$  are obtained from the MRTNSE for the outgoing interface.

Once the resultant value of  $Q_{delay}$  has been substituted into equations (1) and (2) along with the other mentioned parameters, a value of  $R$  to be installed at the outgoing interface is obtained.

Regardless of the reservation class, that is Controlled-Load or Guaranteed Service, if processing of the RES does not result in either the installation of a new reservation or a modification of an existing reservation (i.e. the RES packet was simply a refresh) then the soft-state timer for the reservation is simply reset. Otherwise, the reservation request is propagated immediately down the distribution tree after updating the appropriate fields in the packets header as follows. The end2end delay field in the RES packet is increased by adding to it the following:

- The propagation delay,  $d_{next}$  for the next hop
- An estimate of the current local queuing delay (for the relevant outgoing interface) for data packets of the flow to which the RES packet refers.

In addition, if  $CRTs=11$  the following updates must be made to the RES packet:

- Add the following to the accumulated-bound field of the copied packet.
  1. The propagation delay,  $d_{next}$  for the next hop
  2. The installed local queueing delay bound (for the relevant outgoing interface) for data packets of the flow to which the RES packet refers. This local queueing delay bound is obtained by inserting the reserved value of  $R$  into equation (4) along with the local values of  $C$  and  $D$

$$Q_{local} = \frac{C}{R} + D. \quad (4)$$

- If reshaping to the sender  $T_{spec}$  is being performed at the outgoing interface  
     set  $C_{sum}=D_{sum}=0$ .  
   Else  
     Add the local value of  $C$  to the  $C_{sum}$  field  
     Add the local value of  $D$  to the  $D_{sum}$  field

Once updating of the fields is complete the timestamp field is now set equal to the local clock before forwarding the RES packet to each next hop down the routing tree.

## 5 RETURN(RTN) MESSAGE

The IP destination address of the IP datagram encapsulating an RTN message is equal to the IP address of a previous hop node, the identity of which is obtained from installed Phop state obtained from RES messages, while the IP source address is equal to the IP address of the node out of which the RTN message was sent.

### 5.1 Common part

- **Session** – as for RES message.
- **Nhop** - (object defined in the RSVP protocol) – the identity of the DRP-capable logical outgoing interface that sent this message. The Nhop object consists of the pair (IP address, logical interface handle)
- **Sender address** – the combination of this field and the session object identify a source-based tree. In the case of a shared tree this field is ignored and should be set to all 0's.
- **timestamp** - stamped with the time of the local node clock just before being sent to previous hop up the distribution tree. This is used in calculation of dnext as described in (White 1998).
- **CRTtr** (2 bits) - indicates the receiver's ceiling reservation class.
- **timedelta** - used in calculation of dnext as described in (White 1998).
- **proppdelay** - the data packet propagation delay along the maximum 'Total Rate-Independent Delay'(TRID) path<sup>14</sup> between the node incoming<sup>15</sup> interface out of which the RTN packet was sent and each receiver downstream.
- **pathMTU** - the minimum pathMTU value between the incoming interface out of which the RTN packet was sent and each receiver downstream of that incoming interface.
- **Ctot** - the maximum accumulated Ctot value along the paths between the incoming interface out of which the RTN packet was sent and each receiver downstream of that incoming interface. The C error term is defined in the Guaranteed Service specification (Schenker, 1997).

---

<sup>14</sup> Total Rate Independent Delay(TRID) is given by the sum of the link propagation delays and the D error terms.

<sup>15</sup> The term 'incoming' refers to the direction of data flow. RTN packets are reverse-routed up the distribution tree in the opposite direction to the data flow and so are always sent out of so-called incoming interfaces.

- **Dtot** – sum of D error terms along the maximum ‘Total Rate-Independent Delay(TRID)’ path<sup>14</sup> between the node incoming<sup>16</sup> interface out of which the RTN packet was sent and each receiver downstream. The D error term is defined in the Guaranteed Service specification (Schenker, 1997).
- **path bandwidth** - the maximum path bandwidth value along the paths between the incoming interface out of which the RTN packet was sent and each receiver downstream of that incoming interface.

## 5.2 RTN Guaranteed Service feedback object

The RTN packet may optionally contain a Guaranteed Service feedback object comprising:

- **excess delay field** – the amount by which the installed end-to-end delay bound currently exceeds the target end-to-end delay bound.
- **bottleneck flag** - if set to 1 this indicates that the RTN message has travelled at least as far as the router where the accumulated delay-bound first exceeded the target delay-bound on the first pass of the RES message.
- **Sender Template** – same as that in RES packet whose end-to-end delay bound was exceeded.

## 5.3 RTN state and message merging rules

At an outgoing interface,  $i$  of a router on the distribution tree, reception of an RTN packet from a next hop,  $j$  results in the updating of any matching router state, known as an RTN state entry or  $RTNSE_{ij}$ , or the setup of new state if no match exists. There will be a separate  $RTNSE_{ij}$  for each 4-tuple (Session, sender address, next hop, outgoing logical interface). The first three parameters of this 4-tuple are contained within the received RTN message while the outgoing logical interface(oif) is determined by the interface on which the RTN message arrived. In the case of a shared tree the sender address field will be omitted for the  $RTNSE_{ij}$ . The format of an  $RTNSE_{ij}$  (excluding any guaranteed service feedback parameters) is as shown in Table 2. In addition, for each outgoing logical interface,  $i$  a single Merged RTN State Entry ( $MRTNSE_i$ ) is created from the set of entries  $\{RTNSE_{ij}\}$  for that logical outgoing interface. There will be multiple  $RTNSE_{ij}$ s for a given logical outgoing interface if the logical outgoing interface has multiple next hops on the distribution tree which can occur if the logical outgoing interface connects to a shared medium LAN(e.g. Ethernet). The parameters of the  $MRTNSE_i$  and how they are formed from  $\{RTNSE_{ij}\}$  are also shown in Table 2. The ‘merged values’ of various parameters in each RTN message sent out of an incoming interface to a previous hop upstream are obtained from  $\{MRTNSE_i\}$ , the set of  $MRTNSE$  for the outgoing interfaces as shown in Table 2.

---

<sup>16</sup> The term ‘incoming’ refers to the direction of data flow. RTN packets are reverse-routed up the distribution tree in the opposite direction to the data flow and so are always sent out of so-called incoming interfaces.

$RTNSE_{ij}$	$MRTNSE_i$	<i>Merged RTN packet sent upstream out of interface k</i>
$CRT_{ij}$	$CRT_i = \text{MAX}\{CRT_{ij}\}$	$CRT_k = \text{MAX}\{CRT_i\}$
$Ctot_{ij}$	$Ctot_i = \text{MAX}\{Ctot_{ij}\}$	$Ctot_k = \text{MAX}\{Ctot_i + Clocal_{ki}\}$ (footnote 17)
$Dtot_{ij}$	$Dtot_i = Dtot_{ij}$ Where j is such that $TRID_i = TRID_{ij}$	$Dtot_k = Dtot_i + Dlocal_{ki}$ such that i gives $\text{MAX}\{Dlocal_{ki} + TRID_i\}$ for that interface k (footnote 17)
$Propdelay_{ij}$	$Propdelay_i = \text{propdelay}_{ij}$ Where j is such that $TRID_i = TRID_{ij}$	$Propdelay_k = \text{propdelay}_i + dnext_i$ such that i gives $\text{MAX}\{Dlocal_{ki} + TRID_i\}$ for that interface k (footnote 17)
$PathBandwidth_{ij}$	$PathBandwidth_i = \text{MAX}\{PathBandwidth_{ij}\}$	$PathBandwidth_k = \text{MAX}\{\text{MIN}(\text{pathbandwidth}_i, \text{link rate}_i)\}$
$PathMTU_{ij}$	$PathMTU_i = \text{MIN}\{\text{pathMTU}_{ij}\}$	$PathMTU_k = \text{MIN}\{\text{MIN}(\text{path MTU}_i, \text{linkMTU}_i)\}$
$dnext_{ij}$	$dnext_i = dnext_{ij}$ where j is such that $TRID_i = TRID_{ij}$	
$TRID_{ij} = Dtot_{ij} + \text{propdelay}_{ij} + dnext_{ij}$	$TRID_i = \text{MAX}\{TRID_{ij}\}$	
<i>sender template<sub>s</sub></i>	<i>sender template<sub>s</sub></i>	<i>sender template<sub>s</sub></i>
<i>excessDelay<sub>sj</sub></i>	<i>excess delay<sub>s</sub> = MAX{excessDelay<sub>sj</sub>}</i>	<i>excess delay<sub>s</sub> = MAX{excessDelay<sub>s</sub> - delayReduction<sub>sj</sub>}</i>
<i>bottleneckFlag<sub>sj</sub></i>	<i>bottleneckFlag<sub>s</sub> = MAX{bottleneckFlag<sub>sj</sub>}</i>	<i>bottleneck flag<sub>s</sub> = MAX{bottleneck flag<sub>s</sub>}</i>

Table 2: relationship between RTN state entries, MRTN state entries and merged RTN packets.

<sup>17</sup>  $Clocal_{ki}$ ,  $Dlocal_{ki}$  =router's value of C and D error terms between incoming interface k and outgoing interface i

The last three rows of Table 2 represent optional GS-feedback objects and are written in italics to differentiate them from the core entries shown in the table. Merging between GS-feedback object state only occurs if the objects relate to the same sender template, *s*. A merged GS-feedback object for sender template, *s* is only included in the merged RTN packet sent upstream if the RTN packet is addressed to Phop for sender template *s* as obtained from installed RES state. With regard to the excess delay entries shown in the table, *delayReduction<sub>i</sub>* refers to the local reservation queuing delay reduction achieved since the RES for sender *s* at interface *i* was installed.

If CRT<sub>r</sub> is not equal to GS in the propagated RTN message, the rules in Table 2 are overridden by setting  $Ctot=Dtot=propdelay=0$  in order to ensure that only those links receiving Guaranteed Service are taken into account when conducting worst-case merging of GS-specific parameters.

Whenever the contents of a RTN message to be sent upstream differ from the preceding one, the RTN message is sent immediately. Otherwise, i.e. in the steady state, an RTN message is sent to a previous hop once per some refresh period.

## 6 SUMMARY

In this paper we have discussed the need for resource reservation in the Internet and examined the use of RSVP for this purpose while highlighting some of its favourable characteristics such as its use of 'soft-state' reservations. Consequently we acknowledge RSVP as a useful starting point in the design of alternative reservation protocols but we do not accept that it represents the ultimate solution because of certain deficiencies and restrictions that we demonstrated in the text. This has motivated our design of an alternative IP reservation protocol, DRP which incorporates many principles of RSVP together with the dynamic sender-initiated reservation concept of ABT/IT to achieve the following main goals:

1. High reservation control dynamics to achieve efficient bandwidth usage.
2. Scalability of router-state with regard to number of senders and receivers. The protocol is especially suited to large-scale-multicast applications where it can expect to achieve a router state saving of several orders of magnitude compared to RSVP.
3. Heterogeneity of QoS classes and reservation styles for nodes within a given multicast session.

Details of control messages were presented along with associated processing rules. Although in principle DRP offers considerable benefits over existing reservation protocols certain aspects of it are not well understood and further work is required especially in the following areas:

1. Reservation setup time for each of the different service classes.
2. Impact of SSR mode on reservation set up time compared to SS mode.
3. Effect of worst-case merging of OPWA data – For a large multicast tree this will tend to cause the nodes closest to the sender to over-estimate their local reservations which as a result causes a reduction in the local reservations downstream. Any implications of this phenomenon need to be clarified.

4. Investigation into alternative Guaranteed Service feedback techniques for the purpose of reducing the end-to-end delay bound when it is in excess of the target-delay bound after one pass of the RES packet. For example one alternative worth investigating is the generation of the RTN packet containing the Guaranteed Service feedback object as soon as the bottleneck node is encountered rather than waiting until the RES packet arrives at the receiver.

## 7 ACKNOWLEDGEMENTS

We would like to thank British Telecom Labs, Ipswich, England for supporting this work. In particular we are grateful to Alan O'Neill and Terry Hodgkinson of BT labs for many valuable discussions regarding the material presented in this paper.

## 8 REFERENCES

- Almesberger, W, Boudec, J and Ferrari, T. (1997) Scalable Resource Reservation for the Internet, EPFL DI-LRC, CH-105 Lausanne, Switzerland.
- ATM Forum (1996). ATM User Network Interface (UNI) Specification Version 4.0. AF-UNI-4.0.
- Bilhartz, T., Cain, J., Farrey-Goudreau, E., Fieg, D. and Batsell, S. (1997) Performance and Resource Cost Comparisons for the CBT and PIM Multicast Routing Protocols in DIS Environments, IEEE Journal of Selected Areas in Communications, April 1997. <http://www.epm.ornl.gov/~sgb/pubs.html>.
- Braden, R., Zhang, L., Berson, S., Herzog, S. and Jamin, S. (1996) Resource Reservation Protocol (RSVP) - Version 1 Functional Specification, August 12, 1996.
- Delgrossi, L. and Berger, L. (1995) Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+, August 1995, RFC1819.
- Forge, J., (1979) "ST - A Proposed Internet Stream Protocol", IEN 119, M.I.T. Lincoln Laboratory, 7 September 1979.
- intserv, IETF, Integrated Services Charter (1998) <http://www.ietf.org/html.charters/intserv-charter.html>
- issl, IETF Integrated Services over Specific Link Layers Charter (1998) <http://www.ietf.org/html.charters/issl-charter.html>.
- ITU-T (1995). Q.2931: Broadband Integrated Services Digital Network (B-ISDN); Digital Subscriber Signalling System No. 2 (DSS2); User-Network Interface (UNI) Layer 3 Specification for Basic Call/Connection Control.
- ITU-T (1996) Recommendation I.371. Traffic Control and Congestion Control in B-ISDN, (08/96).
- Schenker, S., C.Partridge, R.Guerin. (1997) Specification of Guaranteed Quality of Service, Request for Comments, September 1997, RFC2212.
- Seidensticker, S., Smith, W. and Myjack, M. (1997) Scenarios and Appropriate Protocols for Distributed Interactive Simulation, Internet Draft, March 1997, draft-ietf-lsma-scenarios-01.txt.
- Topolcic, C. (1990) Experimental Internet Stream Protocol, Version 2 (ST-II), October 1990, RFC1190.



- White, P. (1997) RSVP and Integrated Services in the Internet: a tutorial, IEEE Communications magazine, May 1997.
- White, P. (1998) A case for Dynamic Sender Based Reservations in the Internet. UCL report. <ftp://cs.ucl.ac.uk/darpa/SenRes.ps.Z>.
- Wroclawski, J. (1997) Specification of the Controlled-Load Network Element Service, Request for Comments, September 1997, RFC2211.

## 9 BIOGRAPHY

Paul White was awarded a BEng degree(First Class Honours) in Electronic and Electrical Engineering at the University of Birmingham, England in 1989. From then until 1994 he worked in various fields of hardware/software technology including telecommunications and became a Chartered Electrical Engineer. In 1994 he returned to academia and was awarded an MSc degree(with Distinction) in Data Telecommunications and Networks at the University of Salford, England in 1995. He has been working towards a PhD at University College London since 1995 and is supported by British Telecom Labs, Ipswich, England.

# USD: Scalable Bandwidth Allocation for the Internet

*Zheng Wang*

*Bell Laboratories, Lucent Technologies*

*101 Crawfords Corner Road, Holmdel, NJ 07733*

*zhwang@dnrc.bell-labs.com*

## **Abstract**

In this paper, we present a differentiated service scheme called “User-Share Differentiation (USD)”. The USD scheme is designed for long-term bandwidth allocation without per-session signaling. The scheme allows ISPs to provide traffic isolation on a per-user basis and guarantee proportional fairness. We first look at the background for differentiated services, and the problems with the current proposals. We then present the details of the USD scheme and examine the implementation and deployment issues.

## **Keywords**

Quality of services, differentiated service, weighted fair queuing, bandwidth allocation, proportional fairness

## **INTRODUCTION**

The current Internet is built on the best-effort model where all packets are treated as independent datagrams and are serviced on the FIFO basis. The best effort model does not provide any form of traffic isolation inside the network and the network resources are completely shared by all users. As a result, the Internet suffers from the “Problem of Commons” where greedy users try to grab as much resource as possible. Such a system can become unstable and lead to congestion collapse. The Internet currently still works because most end systems use TCP congestion control mechanisms and back off during congestion. However, such dependence on the end systems’ cooperation is increasingly becoming unrealistic. Inevitably, people start to exploit the weakness of the best effort model to gain more resources. An example of this is to establish multiple TCP connections in web browsers to gain greater share of the bandwidth. The best effort model also prevents ISPs from meeting the different needs of their customers since it is difficult to allocate more resources to those who are willing to pay more.

The problems with the best effort model have been long recognized. For the last a couple of years, QoS provision has been one of the hottest areas in networking research, and various aspects of the issue have been extensively studied including traffic analysis, admission control, resource reservation, scheduling, QoS routing, and operating system support. The architectures of various proposed solutions differ in details. Nevertheless, the underlying model is rather similar. Essentially, applications make resource reservation on an end-to-end per session basis. We refer to this model as the End-to-End Per Session (EEPS) model. In the Internet community, RSVP and INT-SERV are examples of protocols and service models based on this model (Zhang *et al* 1993). In general, the EEPS model achieves QoS guarantees through the following steps:

- The application characterizes its traffic, and describes its requirements in a flow specification.
- The QoS routing figures out one or more candidate paths based on the requirements.
- A reservation or signaling protocol then checks for admission control hop-by-hop and installs the reservation over the candidate path if there is sufficient resources.
- The schedulers enforced the reservation for each flow.

The past work based on the EEPS model has given us valuable insights and practical experience with resource allocation in the Internet. However, the EEPS model has a number of problems:

- On-demand per-session reservation does not work well for Web-based applications. For applications with long lasting sessions such as video conferencing, the delay and overheads of reservation is minimal. But for transaction-based applications such as Web, where a user can go through many destinations in a few seconds, setting up a reservation for each transaction has a high overhead. Furthermore, the resource requirement for Web traffic is usually difficult to determine. Very often, a user does not know how big an object is before fetching it. Also, delay variation affects Web-based application far less drastically compared with applications like video conferencing, thus there is more space for adaptation.
- Security, accounting and administrative support represent a significant amount of overheads in resource reservation. As each reservation is a service contract between the user and ISPs along the path, resource reservation goes far beyond simply installing state inside the network. Each request has to be authenticated and the user's account to be charged. Within the user organization, there may be internal procedures for approval and coordination of requests from individual users. All those have two implications. First, accounting and administrative support has to be an integrated part of a resource reservation system before the system can be deployed. Second, there is a need for aggregated reservation in order to reduce overheads of resource reservation for short-lived sessions.

- The inter-ISP settlement is a complex issue and is unlikely to be resolved in the near term. When a path traverses multiple ISPs, an end-to-end reservation requires an agreement among all major ISPs on the inter-ISP settlement. Any single ISP that does not participate in such an agreement may break the reservation. Incremental deployment measures are essentially for the success of any reservation systems.
- The fine granularity of the EEPS model can lead to some scalability problems as the number of reservations increase. The classifier has to check the five fields (source address, destination address, source port, destination port and protocol) to determine if a packet belongs to one of the reserved flows. Such fine granularity lookup can be expensive when the number of flows is large. When the sessions are short-lived, the control messages can also be substantial and the processing of the control messages may become the bottleneck.

At the time when the work on the EEPS model started a couple of years back, real-time applications such as video conferencing were regarded as the mainstream application for the future Internet. For such applications, the on-demand per-session reservation makes sense. However, the advent of the Web has changed the landscape significantly. The majority of the Internet traffic today is web-based and tends to be short-lived and transaction-oriented.

In the recent months, the term “differentiated services” has been used to describe new service models and mechanisms to achieve bandwidth allocation for aggregated traffic without per-session reservation. The basic requirements for such new service models and mechanisms are as follows:

- Aggregated bandwidth allocation without the need for per-session signalling.
- Long-term service contracts within a single domain.
- Integrated and simplified accounting.
- Better traffic isolation for performance predictability.
- Better services to users who are willing to pay more.

In this paper, we present a scalable differentiated service scheme called “User-Share Differentiation (USD)” (Wang 1997). The USD scheme is designed for long-term bandwidth allocation without per-session signaling. The scheme allows ISPs to provide traffic isolation on a per-user basis and guarantee proportional fairness. We first look at the background for differentiated services, and the problems with the current proposals. We then present the details of the USD scheme and examine the implementation and deployment issues.

## RELATED PROPOSALS

In this section, we examine two related proposals that have been put forward for differentiated services.

### *Premium Service*

The premium service model described by Nichols *et al* (1997) provides guaranteed peak-rate bandwidth for aggregated traffic flows from users at the ISP entry points. The proposal creates a “premium” service that is provisioned according to the worst-case requirements and guaranteed by priority queuing. Routers at the edges of the network filter packets and set the premium bit in the packet header according to users’ premium bandwidth profile. Inside the network, packets with the premium bit set are transmitted prior to the best effort packets.

The premium service proposal represents an extreme form of resource allocation, where the network capacity is effectively reduced by the amount allocated to the premium class traffic and the best effort traffic suffers all the consequences of congestion. The premium class requires strict admission control at the entry points to the ISP. Users’ traffic is shaped to the allocated bandwidth. The arriving packets that exceed the allocated bandwidth profile are either delayed or dropped.

### *Assured Service*

The assured service defined in the profile-based tagging scheme uses drop priority to differentiate traffic (Clark and Wroclawski 1997). Each user is assigned with a service profile that describes the “expected capacity” from the ISP. The traffic from a user is checked by a profile meter at the entry points to the ISP. Packets that are out of the profile can still go through but they are tagged as such by the profile meter. When congestion occurs inside the network, the routers drop the tagged packets first. When traffic is complaint to the agreed profile, it is expected that a user can have predictable level of services.

The premium service proposal and the profile-based tagging proposal are similar in that both proposals create an “upper” class and give preference to the upper class over the best effort class. In both proposals, the classification of packets is done at the edges and the class information is encoded in the packet header. The difference lies in that the premium service drops packets that are out-of-profile at the entry points to the ISP while the assured service still allows such packets go through in the hope that they may get to their destinations.

The two proposals attempt to push all policy-related processing to the edges of the network, and inside the core, the routers just forward packets based on the ToS bits in the header. While this approach has the advantage of simplicity, there are also a number of problems:

- **Provisioning.** Both proposals attempt to provide guaranteed bandwidth to the upper class traffic through admission control at the edges of the network. The assumption here is that with admission control around the edges and proper provisioning of the network, one can effectively eliminates congestion for the upper class. However, given a set of upper class users, the problem of dimensioning the network to meet the bandwidth guarantees to upper class users is a non-trivial problem. Since traffic flows are dynamic; any source can generate traffic to different destinations at different rates and the routes to the destinations may also change. Thus it is difficult for the edges to have the knowledge of traffic distribution inside the network. To provide any sort of

guarantees, it is necessary to provision the network for the worst-case where the one assume that all upper class traffic may go through the weakest link in the network.

- **Choosing profile.** Choosing a proper profile for a user is not a straightforward task either. It can be viewed as the reverse problem of provisioning. For a given network, how can one decide the best profile users can be assigned to. Note that a profile applies to the aggregated traffic flow from a user organization going through the entry point to its ISP. When a network does not have uniform bandwidth provisioning, profiles are likely to be destination-specific. For example, suppose that an ISP has a link to a neighbor ISP with a capacity 600 Mbps and a link to the Internet backbone with a capacity of 100 Mbps. If a user is communicating with another user in the neighbor ISP, the user can have a rate limit of 6 Mbps but only 1 Mbps if the user's traffic goes through the backbone access link. In this case, a profile of either 6 Mbps or 1 Mbps is not appropriate for the user. When a user is sending traffic to multiple destinations, the situation becomes even more complicated.
- **Reverse traffic.** Both the premium service and assured service proposals largely focus on the case where the users send packets towards the ISP. The bits in the packet header are set at the entry points to the ISP before mixed with packets from other sources, therefore the profile meter only need to know the admission control policy for the sender. However, in many cases, the users actually pulling the traffic in from the ISP. A typical example of this is Web-based applications which usually retrieve information from the Web servers. To apply the premium service and assured service models to such reverse traffic, the premium/assured bits have to be set at the server side or at the ISP-ISP boundary. There are a number of problems. First, it implies that the profile meter at an ISP-ISP boundary has to know the admission control policy for all its users. Second, if there may be multiple ISP-ISP boundaries, it becomes necessary for the profile meters at all boundaries to cooperate in order to make sure the sum of all upper class traffic for a user matches its profile.
- **Starvation.** With premium class scheme, that the congestion is invisible to the premium class, the network will no longer to provide any congestion signal for the premium class traffic. For TCP flows in the premium class, the sender's window will grow to the point that all bandwidth allocated to the premium class is taken up. If the bandwidth provisioning for the premium class is not done with care, best effort traffic will see significant degradation and may be starved completely.
- **The profile-based tagging only provides limited protection against misbehaving sources.** Since in profile-based tagging, the network deals with the tagged packets in a FIFO fashion. A misbehaving source can still gain more bandwidth by injecting excessive traffic. The problem can be aggravated when the fixed profiles are significantly over (or below) the level appropriate for the congested links. In such a case, the majority of the packets are not tagged (or tagged). Thus the tagging provides little information to

enforce differentiation, and network behavior will be close to simple FIFO best effort.

## USER-SHARE DIFFERENTIATION

In this section, we present User-Share Differentiation (USD), a scalable bandwidth allocation scheme for differentiated services, and discuss the key design principles behind the scheme.

### 3.1. Overview

The USD scheme is designed to provide long-term bandwidth allocation for aggregated traffic flows. It differs from the premium service and assured service in a number of ways:

- The USD scheme provides traffic isolation between the customers of an ISP rather than between two classes.
- The primary service model in USD is proportional fairness rather than explicit bandwidth guarantee.
- The USD scheme enforces bandwidth allocation on the bottlenecks where the congestion takes place.
- The USD scheme does not require admission control at the edges of the network and it also works well with reverse traffic.

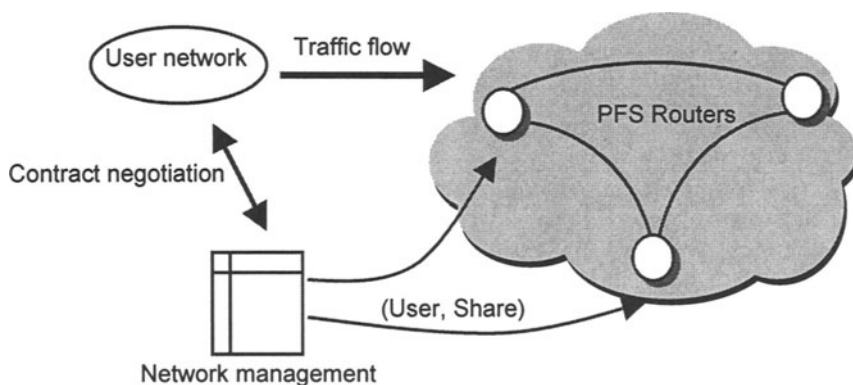
The USD scheme has two important components:

- **User.** A user is the basic entity to which the bandwidth is allocated. The term “user” refers to the party with whom an ISP enters a service contract and the entity that pays for the ISP’s service. It is important to note that a user is not necessarily an end user; it can be a network or a group of networks.
- Each user is assigned a number called “share” based on how much a user has paid for the service. The share is used for determining how much bandwidth a user is allocated to. Under congestion, the share is used as the “weight” in the allocation of bandwidth.

The USD bandwidth allocation is carried out in the following steps:

- At the time when a user subscribes to its ISP for Internet services, the ISP and the user agree on the share for the user based on the user’s requirements and how much it is willing to pay. The share may change each time when the user changes its service contract.
- At any time instance and any point inside the ISP’s domain, the ISP can provide two guarantees to the user. First, a user will have a minimum amount of guaranteed bandwidth anywhere inside the ISP (the worst-case guarantee). And at any time instance, the amount of bandwidth allocated to a user is

proportional to its share among all the active competing traffic on any links (proportional guarantee).



**Figure 1:** User-Share-Differentiation Information Flows

- The user and its corresponding share are distributed to some or all routers inside the ISP through some network management protocol such as SNMP or other similar protocols.
- The USD allocation is policed with a scheduler that supports proportional fair sharing (PFS) and is activated whenever a queue builds up.

We now discuss the key design decisions behind the USD scheme.

### 3.2. Flexible Control Granularity

One of the main issues in any resource allocation is the granularity of the control. The finer granularity offers better control of resources but also brings the associated complexity in the setup of the state and classification of packets. Part of the problem with the EEPS model is its 5-tuple fine granularity and the signaling requirement that comes with it. On the other hand, the granularity decides the minimum level of control one can excise, and the level of traffic isolation can be supported. Therefore, it is an important engineering decision that has to be made.

In USD, we follow the natural administrative boundaries in the customer-ISP contractual relationship and introduce the user as the basic unit that defines control granularity. All traffic originated from or destined to a user is aggregated into a single flow, and the ISP provides protection for a user's traffic from other competing users. Within traffic of a single user, it is up to the user to decide how the bandwidth is used internally. The definition of user is flexible to allow variable granularity to meet different requirements. A user is an individual host identified by its IP address or it can also be a network identified by the network prefix, and an ISP identified by its prefixes.



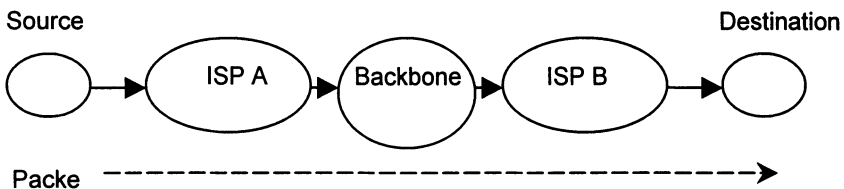
As the per-user granularity provides full traffic isolation between users, it takes away the incentives for misbehaving. If a misbehaving user ignores the congestion signal, and continues to send traffic at unsustainable rates, it can only waste the bandwidth that the user is allocated and cause its own packets to be dropped. We believe that once the traffic isolation is provided inside the network, users will start to deploy intelligent control congestion mechanisms for their own good.

### 3.3. Scalable Aggregation

The definition of user also determines the level of aggregation inside a network. Note that the Internet has a hierarchical structure of ISPs, from backbone ISPs to retail ISPs and to end users. USD follows the same structure and allows hierarchical aggregation.

When traffic goes across ISP boundaries, the level of traffic aggregation also changes accordingly. Within a user's immediate ISP to which the user has a direct contractual agreement, all traffic of the user is aggregated into a single flow. In the core backbone, the retail ISP has a contractual agreement with the backbone provider on the behalf of all users from the retail ISP. Thus, all traffic from and to the ISP is visible in the backbone as one flow. As the traffic moves from the sender toward the backbone, the level of aggregation increases while the level of control granularity decreases. Such variable levels of aggregation is essential for the scalability in the core backbone as the amount of state in a network only depends on the number of customers an ISP has direct contractual relationship.

When a packet traverses the network, the control policy can actually change as the user-ISP relationship changes. Take Fig. 2 for example. When a packet from user A enters ISP A, the packet is aggregated into the flow to or from user A and the bandwidth allocation within ISP A is determined by the share assigned to user A (the source address prefix). When the packet goes into ISP B, the destination address prefix becomes visible thus the allocation depends on the user B's contract with ISP B. Within the backbone, the packet is aggregated into the flow for ISP A or ISP B. Such variable level of aggregation ensures a great deal of scalability and is consistent with the contractual obligations for the parties along the path.



**Figure 2:** Variable Levels of Aggregation

## 1.4. Proportional Fair Sharing

The per-user granularity allows traffic isolation between competing users. We can now move onto the policy for allocating bandwidth to multiple competing users.

In a commercial Internet, it is natural that the bandwidth allocation must be linked to how much a user has paid. This way, users can quantify the amount of services they get and thus are willing to pay more for better services. There are two basic approaches to achieve this goal. One can provide a user explicit bandwidth guarantee. Such guarantee would be easily to carry out if it is over a specific path, for example, a virtual leased line between two sites for a VPN application. However, to guarantee bandwidth anywhere in an ISP is much harder as one has to provision for the worst-case scenario. We believe that, for long-term bandwidth allocation anywhere in an ISP, guarantees for the relative fairness rather than the explicit amount of bandwidth is a more efficient option. In USD, the share reflects how big the slice of service that a user has paid for. When congestion occurs, the USD scheme allocates bandwidth to all active users at the bottleneck in proportional to the shares. For example, if user A and user B have shares of 5 and 10 respectively, user B will always get at least twice of what user A gets. If a user is consuming less bandwidth than it is allocated, the spare bandwidth is allocated in proportional to the other backlogged users. We call such a sharing model as “Proportional Fair Sharing”. As we discuss in the next section, such sharing policy can be easily implemented with WFQ using the share as the weight.

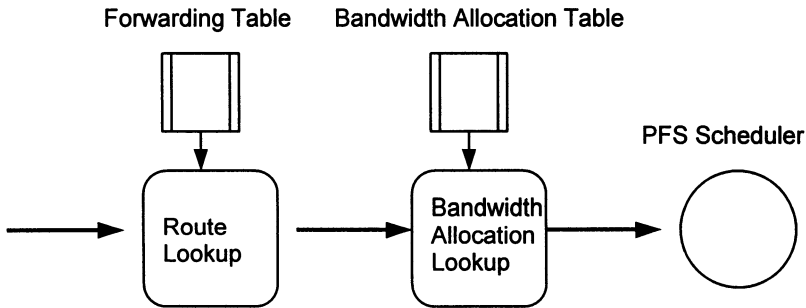
Note that in the worst-case, the proportional allocation yields the same result as explicit allocation. For example, suppose that an ISP has 4 users A, B, C and D, sharing an access link of 30 Mbps. The agreed allocation is 4 Mbps, 6 Mbps, 8 Mbps and 12 Mbps respectively. This allocation can be described with the actual bandwidth, 4 Mbps, 6 Mbps, 8 Mbps and 12 Mbps. Alternatively, the allocation can also be expressed with relative sharing, 2:3:4:6. When the 4 users are all active over the link, the bandwidth allocation is the same. However, the relative sharing has a number of advantages. First of all, it can guarantee the same minimum bandwidth allocation as an explicit allocation does. Second, it allows the bandwidth above the minimum to be shared in proportion to the minimum allocation. For example, suppose that user A and B in the previous example are not using their allocated bandwidth during a period. Now user C and D can share the extra bandwidth in proportion to their relative ratio. The final allocation to user C and D becomes 12 Mbps and 18 Mbps. More importantly, the relative sharing representation works well with multiple bottlenecks with different bandwidth provision. For example, suppose the ISP of 4 users has another link with 600 Mbps bandwidth. The 4 users who have shares of 2, 3, 4, and 6 will have minimum guaranteed bandwidth automatically scaled up to 80 Mbps, 120 Mbps, 160 Mbps and 240 Mbps respectively.

The relative sharing can be viewed as a flexible profile as it scales up and down according to the bandwidth available whilst guaranteeing the minimum bandwidth. In practice, the share can be defined in such a way that the share for a user can be easily derived from the minimum bandwidth allocated. For example, if we define

the unit of share is 1 kbps, a user with 4 Mbps minimum band-width has a share of 4000.

## IMPLEMENTATION AND DEPLOYMENT

Fig. 3 shows the block diagram for the implementation of the USD scheme in a router. The bandwidth allocation unit is similar to the IP lookup unit in IP routers and it consists of a bandwidth allocation table and a table lookup engine. The bandwidth allocation table is a list of user prefix and its associated share. The bandwidth allocation lookup engine does a longest prefix match with the source and destination of each packet to see if there is a match in the resource allocation table. If there is a match for either the source or the destination address, the associated share is used in the scheduler as the weight for bandwidth allocation. If both the source and destination match, this implies that both the sender and the destination are within the ISP. In such cases, the minimum of the two shares will be used for scheduling.



**Figure 3:** Block Diagram for the USD Implementation

To support USD, routers need to implement a scheduler that supports proportional fair sharing. There is a wide range of scheduling algorithms that meet such requirements. For example, Weighted Fair Queuing (WFQ) is such an algorithm that has been extensively in recent years (Parekh and Gallager 1992). Although the original WFQ is expensive to implement, several variations of WFQ have been proposed that support band-width sharing in similar fashion but are optimized for software and hardware implementation (Bennett and Zhang 1996, Shreedhar and Varghese 1995, Stiliadis 1996). Some of the algorithms can emulate WFQ closely with  $O(1)$  complexity (Shreedhar and Varghese 1995).

USD enforces bandwidth sharing locally on the bottleneck links. Thus it does not require any changes to the end systems and any admission control at the user-ISP boundaries. Consequently, USD can be deployed in an incremental fashion. In fact, routers can be upgraded to support USD individually and each upgrade gives incremental improvement to the whole network. For example, when USD is

installed on the router connected to the access link to the backbone, bandwidth allocation is enforced immediately for all traffic that is going through the access link. Moreover, USD only needs to be deployed at the points in the network that are heavily congested. Once bandwidth sharing is enforced at those points, other links may not require further policing.

## CONCLUSIONS

In this paper, we examined the problems with the Premium service and Assured service proposals for the differentiated services, and presented the details of the USD scheme. We conclude that although the USD scheme requires more sophisticated support in the core network routers, it provides both minimum bandwidth guarantees and proportional fair sharing across the network and under various provisioning.

## REFERENCES

- Bennett, J. and Zhang, H (1996) WF2Q: Worst-case fair weighted fair queuing, *IEEE INFOCOM'96*.
- Clark, D., Wroclawski, J. (1997) An Approach to Service Allocation in the Internet, Internet Draft, available at <http://diffserv.lcs.mit.edu/Drafts/draft-clark-diff-svc-alloc.00.txt>.
- Nichols, K., Jacobson, V. and Zhang, L. (1997) A Two-bit Differentiated Services Architecture for the Internet, Internet Draft, available at <http://diffserv.lcs.mit.edu/Drafts/draft-nichols-diff-svc-arch-00.txt>.
- Parekh, A. K. and Gallager, G. R. (1992) A generalized processor sharing approach to flow control - the single node case, *IEEE INFOCOM'92*.
- Shreedhar, M. and Varghese, G. (1995) Efficient Fair Queuing using Deficit Round Robin, *ACM SIGCOMM'95*.
- Stiliadis, D. (1996) Traffic Scheduling in Packet Switched Networks: Analysis, Design and Implementation, Ph.D. Thesis, UC Santa Cruz, Ca, USA.
- Wang, Z. (1997) User-Share Differentiation – scalable service allocation for the Internet, Internet Draft, available at <http://diffserv.lcs.mit.edu/Drafts/draft-wang-diff-serv-usd-00.txt>.
- Zhang, L., Deering, S., Estrin, D., Shenker, S. and Zappala, D. (1993) A New Resource Reservation Protocol, *IEEE Network*, 7 (5): 8-19.

# **A Connectionless Approach to Providing QoS in IP Networks**

*B. Nandy, N. Seddigh, A.S.J. Chapman and J. Hadi Salim*

*Computing Technology Lab, Nortel*

*PO Box 3511, Station C, Ottawa, ON K1Y 4H7, Canada*

*Phone: (613) 765-3709; Fax: (613) 763-8855*

*E-mail: {bnandy, nseddigh, achapman, hadi}@nortel.ca*

## **Abstract**

The attempt to provide QoS in IP networks has raised some interesting questions on how a service can be provided to meet the application requirements while obeying the network resource constraints. Previous efforts focussed on a flow-based, connection oriented approach to deliver QoS for IP Networks - Intserv. This approach was quite comprehensive but it has not been widely deployed because of complexity and scalability issues. A recent packet marking based scheme called Differentiated Services (Diffserv) Architecture provides a relatively simple and coarse approach. It is too early to predict the usefulness of this approach. This paper outlines a framework to deliver IP QoS which is based on Intserv. It addresses scalability concerns by removing the need for a connection-oriented reservation setup mechanism and replaces it with a Diffserv-like mechanism to consistently allocate bandwidth end-to-end in a network. A prototype device is discussed that manages bandwidth on a node. An algorithm is presented that allows the device to automatically detect application QoS requirements without the need for application-level signalling. A priority-based scheduling mechanism with a variant of weighted round-robin is described.

## **Keywords**

**QoS, IP Networks, Connectionless, Diffserv, Intserv**

## 1 INTRODUCTION

The Internet has traditionally offered services in a best-effort manner. This is acceptable in an environment where congestion due to bandwidth requirement is seldom. Moreover, most of the traditional applications like email, file transfer, news are relatively insensitive to delay and delay variations. However, recent surge in Internet popularity has caused the bandwidth to be at premium. The new applications like streaming video have high bandwidth requirements as well as delay constraints. Web access, real-audio etc. require a service which is better than best-effort. Adding more bandwidth is no longer a solution. Thus, a mechanism is needed for bandwidth sharing and prioritization. The idea of prioritizing traffic is also in tune with the approach of commercializing internet applications i.e., to get better service, one needs to pay more.

A mechanism for bandwidth management is needed by which the finite available network resources can be shared among various applications in a manner suitable to the specific applications and also following the guidelines from the administrator.

### 1.1 Providing QoS via Integrated Services (Intserv) Architecture

Initial efforts on providing QoS for IP networks focussed on the Intserv (Integrated Services) model. This model relied on a flow-based, connection-oriented approach to deliver QoS for IP networks. An overview of Intserv is provided in RFC 1633 [1]. In this RFC, Intserv extends the current IP architecture so that it provides QoS to end users. The extension includes: (i) A service model with two services (ii) A reference implementation framework to provide support for QoS-enabled routers.

The network element behaviour required to support the two services are outlined in: (i) RFC 2211: Controlled Load [2] and (ii) RFC 2212: Guaranteed Service [3]. The reference implementation framework provides implementation-level detail to realize the above two services.

The framework [1] proposed to provide QoS support in routers includes the following four elements: (i) classifier, (ii) scheduler, (iii) admission controller, (iv) reservation setup protocol. RSVP is used as the reservation setup protocol of choice [4].

### 1.2 Providing QoS via Differentiated Services (Diffserv) Architecture

The Intserv model is quite mature and much work has been completed. However, it has not been widely deployed due to a variety of concerns. A primary issue is that of scalability [6]. With Intserv, intermediate routers need to save per-flow state information. Another concern is the end-to-end connection-oriented approach of RSVP [4] which is foreign to IP networks. The above issues result in an architecture that is complex to implement and deploy. This might be the single most important reason why the new Differentiated Services initiative has started.

The Differentiated Services Architecture (Diffserv) attempts to address the above concerns by operating on the premise that it is beneficial to move complexity to the

edge of the network and keep the core simple [12]. It intends to provide differing levels of service in the Internet without the need for per-flow state and signaling at each router. The framework includes the following elements: (i) A Traffic Conditioning element at the edge of the network that performs the following: marks packets to receive certain levels of service at the backbone, polices packets and performs traffic shaping to ensure that packets entering the backbone conform to network policies (ii) Core routers that treat packets differently depending on the packet marking completed by the edge device (iii) Allocation/policy mechanism that translates into end-to-end QoS levels of service seen by the end-users [12].

### 1.3 Our Work

This paper discusses a scheme for providing end-to-end QoS which is driven by application requirements. The approach is based on Intserv but does not utilize RSVP, thus addressing the scalability concerns [6] expressed about the Intserv model. Our approach consists of the following: The first element is a device called the Traffic Conditioner which manages the bandwidth at router junction points. The second element is a connectionless mechanism for ensuring consistent end-to-end delivery of QoS based on application requirements.

The Traffic Conditioner is based on the reference implementation framework described in RFC 1633. It contains three of the elements required to manage bandwidth on a particular router junction point: (i) Classifier (ii) Admission Controller (iii) Scheduler. However, instead of RSVP, it utilizes a scheme proposed in [5] to automatically discover QoS requirements for traffic flows and services them accordingly.

In the RSVP model, host machines would initiate connection-setup end-to-end before engaging in data transfer. This connection-setup is used to communicate application QoS requirements and determine if the necessary network resources are available before starting data transfer. This type of pre-negotiation is not intrinsic to IP networks.

The Traffic Conditioner automatically and dynamically meets application requirements without the need for pre-negotiation. Instead, it performs on-the-fly traffic characterization to put network traffic into different classes which are then serviced by a scheduler in such manner that reflects application requirements for that class of traffic.

Having discovered the requirements for a particular flow, and classified the packet accordingly, the Traffic Conditioner marks the packet with its class.

The second element required is a mechanism to ensure consistent allocation of bandwidth across all routers to provide end-to-end QoS. This is an area of ongoing research and is discussed further in Section 3.0. The connectionless approach utilized is similar to the scheme used in the Differentiated Services Architecture outlined earlier.

The paper is organized as follows: Section 2 describes the functional detail of the Traffic Conditioner including the classification, admission control and scheduling

schemes utilized. Section 3 outlines the necessity of consistent bandwidth allocation mechanism. Section 4 presents the experimental results gained from implementing the prototypical Traffic Conditioner. Section 5 contains the conclusion and discusses future plans.

## 2 TRAFFIC CONDITIONER: FUNCTIONAL DETAIL

With the advent of new applications, the traffic pattern in IP networks (Internet) has changed over the years. The present traffic at the IP networks can be broadly divided into three categories in: (i) Real-Time, (ii) Interactive and (iii) Bulk. The real time voice and video requires an uninterrupted data stream which keeps the maximum delay variation within a limit. Video conferencing requires the total delay to be within a limit so that the human interaction is not affected. The traffic generated from Telnet, X-windows and web browsing are also interactive in nature and requires a good response time. Another category is bulk transfer like file transfer and NFS (Network File System) backup which do not have any stringent delay variation requirement. At the same time, this bulk traffic should not cause congestion for other classes of traffics.

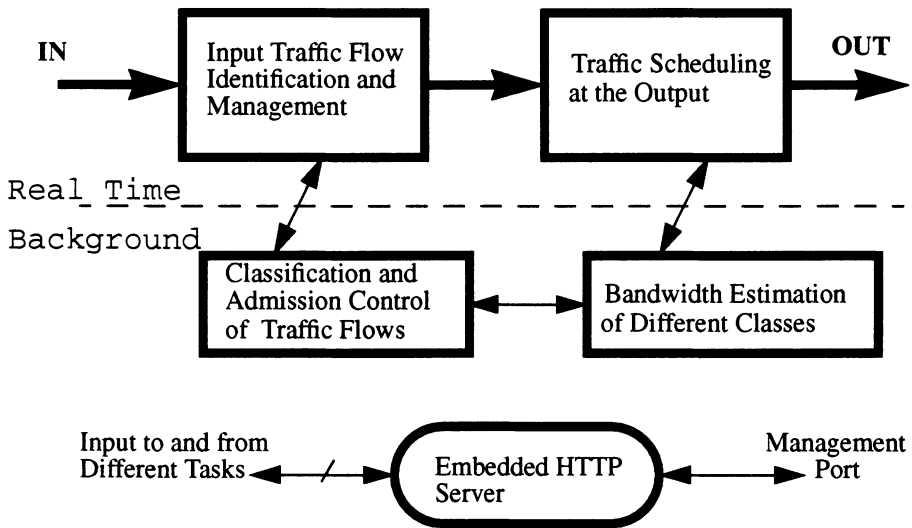
Figure 1 shows a Traffic Conditioner functional model. Traffic Conditioner performs bandwidth management on an individual packet stream (i.e., the packets flowing between two applications in client and server). Each packet belongs to a flow. Flows are uniquely identified by source and destination ip addresses, transport level port pairs and protocol type. Flows are important, since the classification is performed by the characteristics of a stream of packets between a pair of users.

Any packet arriving at the input of the Traffic Conditioner is associated with a flow. All the flows are maintained in a flow-list. If the arrived packet does not belong to any existing flow in the flow-list, a new flow entry is attached to the flow-list. The packet is queued at the scheduling class queue for the flow's class. The scheduler outputs the packets based on the class and the scheduling strategy. The real-time data path (as shown in Figure 1) has two major functions: (i) Identify the flow for the input packet and (ii) schedule the packet to the output. A flow entry to a flow-list is removed if there is no packet arrival to the flow for a fixed time (e.g., two seconds).

The background functions (as shown in Figure 1) are to (i) classify the flows, (ii) admission control and (iii) perform bandwidth estimation of different classes of traffic. The classification of each flow is performed on the fly based on traffic characteristics (e.g., bit rate, packets per second etc.). The classification is performed periodically on all the flows in the flow-list. The packets arriving at the input before the proper flow classification are scheduled with default class.

The bandwidth estimator updates the bandwidth usage on each class periodically. This is necessary for the admission controller and the classifier at the classification time. An embedded HTTP server is provided for monitoring and setting parameters by the administrator.





**FIGURE 1. Traffic Conditioner Functional Blocks**

## 2.1 Flow Classification and Admission Control

The paper [11] proposes a hierarchical link sharing mechanism to address the requirements for realtime traffic in presence of other classes of traffic. The classes in the hierarchical link-sharing structure can be multiple agencies, multiple applications, multiple protocols etc. and the mechanism used for link sharing is called class based queueing (CBQ). The experimental results reported in [9], show that the CBQ based link sharing mechanism between CBR type of traffic with two different priority of TCP traffics. It also shows the delay behavior of the traffics in the shared link.

The Traffic Conditioner classification strategy is based on the scheme proposed by [5]. The flow classification for both TCP and UDP flows are performed on the basis of different treatments required for different traffic types i.e., the application requirements. Thus, different applications with the similar service requirement will fall under the same class. These classes can be different nodes in the hierarchical link-sharing structure as proposed in [11].

The idea behind classifying TCP flows is to separate traffic requiring fast response from the delay insensitive bulk transfer. The UDP flows are classified to differentiate between traffic requiring (i) low latency and low bandwidth, (ii) low latency and high bandwidth and (iii) delay insensitive bulk transfer. The classification of a traffic flow is performed on the basis of traffic characteristics rather than identifying the well known ports (e.g, 80 for web) although that can be used to assist the classification.

The Traffic Conditioner divides traffic into the following classes:

**Interactive:** The TCP flows with short packets and requiring short round trip time are captured in this class. The applications like, Telnet, web browsing and interactive X-windows etc. will fall in this category. The objective here is to protect a portion of the total bandwidth to ensure a reasonable response time. A short packet is defined as a packet with less than or equal to 128 bytes.

All the TCP flows are classified as interactive at the beginning. If the number of continuous long packets exceed a threshold (e.g., 200) without a string of two or more short packets, the flow is moved to bulk transfer class. If the class bandwidth is available, the class is considered as Bulk Transfer with Reserved bandwidth. Otherwise, the flow class is Bulk Transfer with Best Effort.

**Bulk Transfer with Reserved Bandwidth:** The TCP flows with continuous long packets are captured in this class. Applications like, large FTP, Web image transfer will fall in this class. It is ensured that the bulk flows get a certain portion of the total bandwidth and at the same time bulk traffic should not encroach in allocated bandwidth of other classes of traffic.

If there are two or more continuous short packets in the stream of long packets, the flow is reverted back to Interactive class.

**Bulk Transfer with Best Effort:** The TCP flows with continuous long packets but no class bandwidth available at the reserved category falls in this class. This traffic is scheduled on best effort basis.

If bandwidth becomes available, the flow makes a transition to Bulk Transfer with Reserved Bandwidth class. If there are two or more continuous short packets in the stream of long packets, the flow is reverted back to Interactive class.

**Low Latency:** The UDP flows with low packet rates are captured in this class. The applications like real audio, interactive voice, NFS requests and short replies, DNS (Domain Name Server) transactions fall in this category. The objective of this class of traffic is to treat it with high priority so that the latency remains low.

All the new UDP flows are classified as Low Latency at the beginning. The flow is moved to UDP Best Effort if the packet rate exceeds packet rate threshold or no bandwidth is available at Low Latency class.

**Best Effort:** The UDP flows with high packet rate but not classified as Real Time class falls in this class. NFS file backup is one application belongs to this category. The traffic is transferred in a best effort basis.

The flows matching the Real Time template are moved to the Real Time class if that class has available remaining bandwidth. Flows can also have a path back to Low Latency class.

**Real Time:** The UDP flows like streaming video and NFS based video are in this class. The traffic is handled with high priority so that the latency remains at the minimum. The reason for distinguishing Low Latency and Real Time is to preserve the bandwidth for Low Latency class of traffic.

The streaming real time traffic arrives at a constant rate (with a slight variation due to network delay) at the receiver. The distribution of packet interarrival times is

uni-modal. On the contrary, NFS based file transfer, sends a fixed number of packets before it waits for acknowledgment from the application layer. It means that the distribution of packet interarrival time for NFS based application is bi-modal which is different from unimodal distribution of Real Time class of traffic. Also, if the packet rate of the flow in Real Time class exceeds a certain threshold (which makes it as unlikely to be video), the flow is reverted back to UDP Best Effort class. If the flow is idle for more than a second, its class is changed to Low Latency.

As mentioned in [10] to provide bounded delay service, networks must use admission control to regulate the load. In this work, the measurement based admission controller enforces a limit on the flows which require high bandwidth and high priority scheduling. The admission control is enforced on the Real-time flows based on the measurement of link usage. Bandwidth Estimator periodically updates the bandwidth usage of Real-time class. Any new Real-time flow is admitted (i.e., allowed to be serviced) only if the new bandwidth requirement of the class is within the administrator specified limit. Otherwise, the flow is marked as *reject*. Packets of flows belonging to reject class are dropped by the traffic conditioner. An ICMP (Internet Control Message Protocol) host unreachable message is sent back to the source host to stop the flow.

## 2.2 Scheduling

A key component in bandwidth management of a link is a scheduling mechanism. In the work [11], the usefulness of priority scheduling mechanism for link sharing is studied. The scheduling of the classified flows in Traffic Conditioner is performed with two priorities: high and low. The high priority traffic is scheduled without any delay at the scheduling queue and limited by the admission control mechanism. The low priority traffic is scheduled according to a set of rules based on the allocation of bandwidth and a criteria for sharing bandwidth with other classes of traffic on the link. Traffic in the Real Time and Low Latency classes are handled with high priority. Traffic in the Interactive, Bulk Transfer with Reserved Bandwidth and Best Effort classes are scheduled with a low priority. The ideas are: to protect a portion of bandwidth for Interactive class of traffic to guarantee a low round trip time, to limit the delay insensitive Bulk Transfer traffic to a specified limit so that it does not hog bandwidth from other classes. The rule based approach for scheduling of classes with low priority will achieve a similar goal of weighted round robin [11], with weights proportional to the combination of allocated bandwidth per class and delay sensitivity of the class.

A scheduling window of  $T$  seconds (1 sec. for this implementation) is chosen. The bandwidth allocation for each class during  $T$  seconds is proportional to the administrator specified bandwidth values. The window is divided to sub-windows of a smaller  $t$  milli seconds intervals. The scheduler wakes up on every  $t$  seconds and schedules the packets arrived at different class queues during  $t$  seconds with the following rules:

1. All the high priority packets (UDP Low Latency and Real Time) are transmitted until there are no packets remaining in these queues. The queues are served in a round robin fashion.
2. The packets at Interactive and Reserved Bandwidth classes are transmitted only if there is no remaining high-priority packet at the class queue. These two queues are served in a round robin fashion.
3. The packets from Reserved Bandwidth class queue are transmitted only if the allocated bandwidth limitation for the class during scheduling window of T seconds is not exceeded.
4. The packets from Interactive class queue are transmitted even if the allocated bandwidth for the class is exceeded (in a scheduling window of T seconds) but there is no packet waiting at the Best Effort and Reserved Bandwidth classes and bandwidth is available in those two classes.
5. The best-effort class of packets are transmitted if there is no packet in any other class of traffic and bandwidth is available for this class. These packets are also transmitted if its allocated bandwidth is exceeded (in a scheduling window of T seconds) but there are no remaining packets at the Reserved Bandwidth class and bandwidth is available for this class.
6. Packets from Reserved Bandwidth class and Best Effort class are not allowed to borrow bandwidth from Interactive class of traffic. Thus, the interactive bandwidth is preserved.

The scheduling class queue length at the traffic conditioner for different classes of traffic are different. The queue length will never grow for high priority traffic. The queue length does not grow for interactive traffic, since the bandwidth is preserved for this class and allowed to steal bandwidth from other classes. Queue length of Bulk Transfer with Reserved Bandwidth and TCP Best Effort will grow with traffic but TCP adjusts to keep the queue length minimal. The queue length of UDP Best Effort has the possibility of growing if the traffic exceeds its allowed limit. Thus a congestion control mechanism similar to drop tail is implemented to restrict the excessive traffic for this class.

### **3 CONNECTIONLESS APPROACH FOR END-TO-END BANDWIDTH ALLOCATION**

The Traffic Conditioner is one element in the connectionless approach to QoS in IP networks. Ongoing investigation and experimentation is being pursued to study the second component, i.e. providing a mechanism to emulate end-to-end reservation setup. The issue of providing QoS similar to Intserv's Guaranteed Service Model [3] without using an RSVP-like resource reservation mechanism, remains an open research area. It is not clear whether or not the Differentiated Services initiative will be able to provide this service.

However, we believe that it should be possible to achieve the Controlled Load Service model [2] using the approach outlined in this paper. In order to do this a

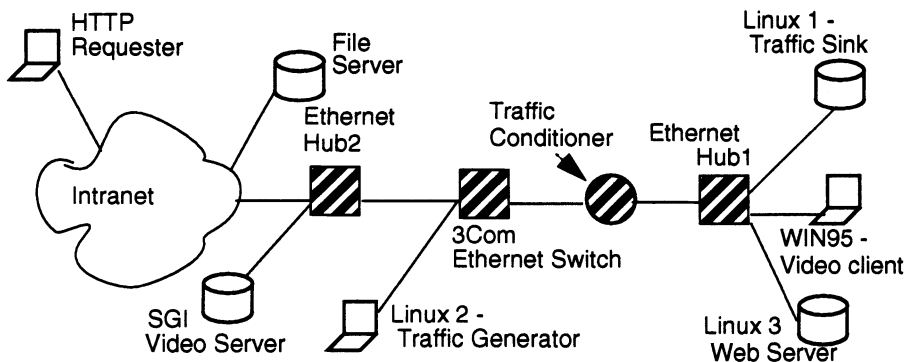
mechanism is required to treat flows in a consistent manner in all routers along the path from source to destination. To avoid the pitfalls of RSVP, this should be achieved without per-flow state saving at each router.

This can be achieved in a well-engineered network with packet marking by the Traffic Conditioner. The administrator statically preallocates bandwidth for each of the defined classes in a consistent manner across the network. This can easily be achieved in an intranet environment. However, in the internet, the solution evolving out of the Diffserv initiative should be applicable. Proposals have been made that discuss dynamic allocation of bandwidth in a connectionless environment [7].

## 4 EXPERIMENTAL RESULTS

The Traffic Conditioner (TC) was implemented on a Pentium 200MHz PC running VxWorks as the RTOS (Real Time Operating System). The current implementation utilizes a 4-port OSICOM PCI ethernet card. One port is used for network management. The other two ports connect to the link that the Traffic Conditioner is conditioning. The TC also contains Mombasa - an embedded web server that is used to facilitate device network management. With the inclusion of Mombasa, network administrators are able to manage the Traffic Conditioner using any standard HTTP web browser. e.g. Netscape or Internet Explorer. Currently, Mombasa is used for configuration and statistics monitoring.

Although this study would have benefitted from a hardware implementation, the software implementation of the TC was able to support traffic levels on a 10Base-T ethernet LAN.

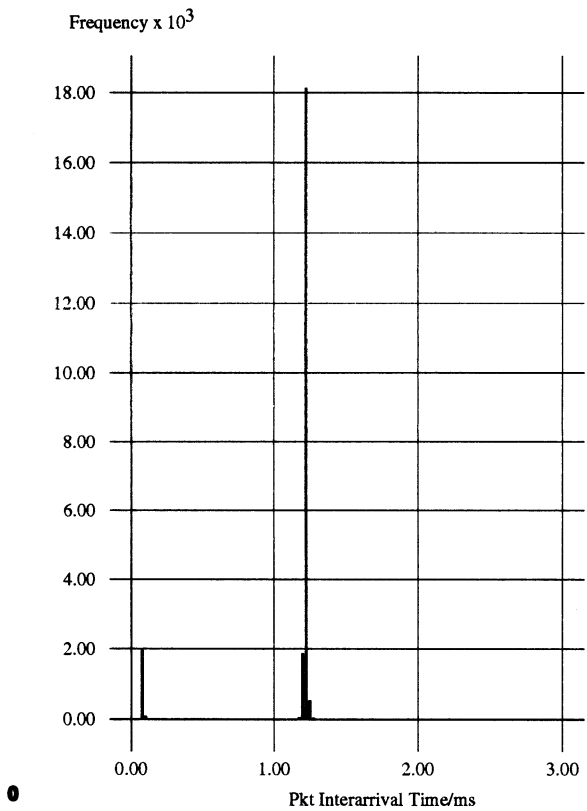


**FIGURE 2. Experimental Setup**

For its operation, the TC sets the ethernet cards in promiscuous and NSAI modes. Promiscuous mode allows it to take in all packets on the line. NSAI mode prevents the ethernet card from stamping its MAC address as the source. This allows the TC to operate transparently on the link.

The implementation model used by the TC closely reflects that depicted in Figure 1 with the Flow management, and Scheduling processes carried out as foreground processes and the classification carried out as a background process.

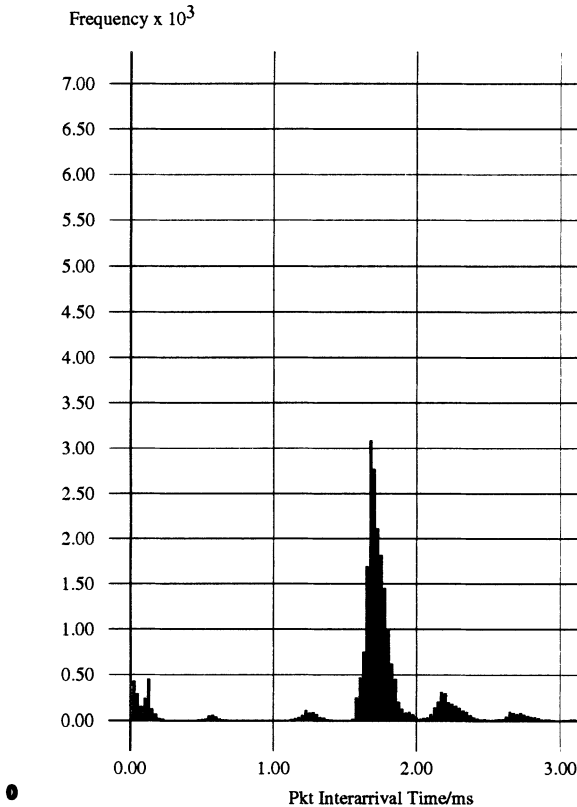
As mentioned previously, RFC 1633 [1] provides a reference implementation model for a QoS-enabled router. The intention of this set of experiments was to study the QoS model elements referred to in that RFC. To do this, the TC was developed as a stand-alone device intended to perform bandwidth management over a single link. The rest of this section describes the results obtained when the Traffic Conditioner was deployed on the network in the presence of various types and rates of traffic



**FIGURE 3. Interarrival Times - without TC; Background Load - 0.5Mbps**

The experimental setup is depicted in Figure 2. As the Figure shows, the TC was deployed in a live network. To carry out the tests, varying levels of network traffic needed to be created. A deterministic traffic generator was used to generate traffic with varying packet lengths and sending rates for both UDP and TCP. As the Figure

2 shows, the setup consisted of a Silicon Graphics video server, a File Server, a Windows 95 computer with streaming video client software, Linux2 which served as the Traffic Generating computer, Linux1 which served as the sink computer for the traffic generator, and Linux 3 which was used as Web server.



**FIGURE 4. Interarrival Times - without TC; Background Load - 6.2 Mbps**

A key outcome of the experiments below was the verification of the classification scheme proposed in [4]. Various types and rates of traffic were generated and it was observed that all fit into expected classes. The experiment was performed with the following applications: ftp, telnet, streaming video, real audio, web browsing (small and large files), DNS and NFS based file transfer.

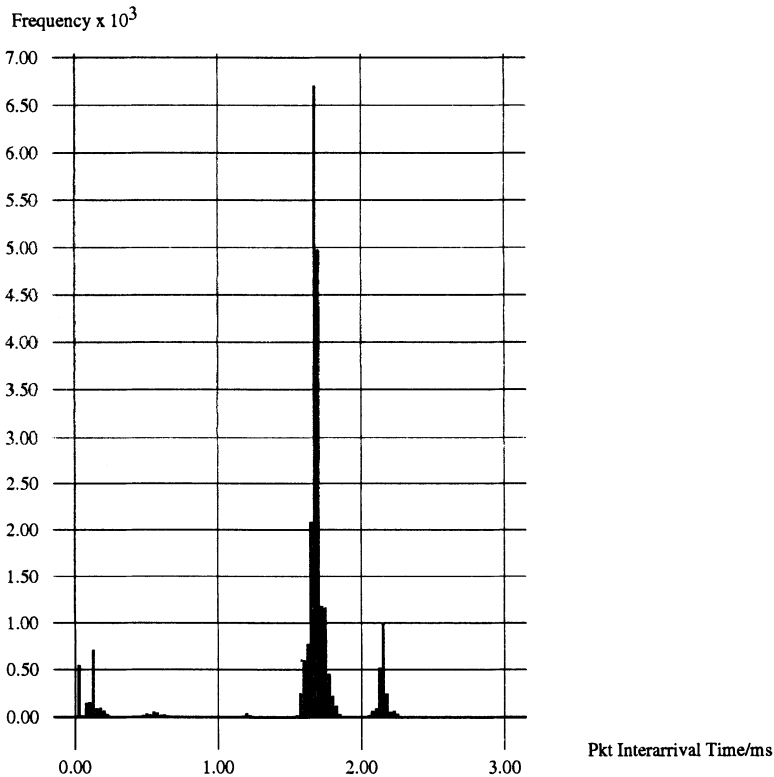
#### **Experiment 1 - Behavior of Video Traffic in presence of network traffic**

The goal of the 1st experiment was to explore how real-time traffic such as video behaved in the presence of network traffic. It was desirable to observe its behavior

in scenarios with and without the TC. Video sessions were started from the Win95 computer. This resulted in around 1.4 Mbps of streaming video being transferred from the SGI video server to the Win 95 video client. Background network traffic was generated with Linux2 as the source and Linux1 as the sink.

Three separate tests were carried out. Packet interarrival times were measured for the video stream in the following test cases:

1. Without Traffic Conditioner; Low network traffic rates
2. Without Traffic Conditioner; High network traffic rates
3. With Traffic Conditioner; High network traffic rates



**FIGURE 5. Interarrival Times with TC; Background Load - 6.2Mbps**

In all cases, the experiment was run for 3 minutes and around 25,000 samples obtained. Measurements were obtained on the TC interface connected to Ethernet Hub1. The results of the three tests can be seen in Figures 3, 4 and 5. Each Figure plots a histogram of the packet interarrival times for the video traffic.

Figure 3 shows the packet interarrival distribution for a network with very low traffic levels. The distribution is a clear reflection of video traffic characteristics.



Two clear peaks can be seen on the distribution. There was a third peak at around 80 ms but that set of data was filtered out for all three Figures as it is not much of comparative value. The peak around 1.25 ms accounted for almost 84% of the 25,000 data samples. The other peaks at 0.1 and 80 ms appeared equal in height and width. This reflects the nature of traffic from this particular video stream which we verified sends packets in 12-packet cycles - 10 packets with a delay of 1.25 ms, 1 packet with a delay of 0.1ms and 1 packet with a delay of 80 ms.

The next testcase involved the same video session but also, introduced a background network traffic load of 6.2 Mbps from the Linux2 to the Linux1 computer. The measurements from this test can be seen in Figure 4. As can be seen from the Figure 4, the height of the main peak reduced greatly and its width expanded considerably. Quantitatively, the width expanded from 0.23 ms to 0.45 ms, almost doubling in size. This kind of effect on video traffic is not desirable as it affects the quality of images that is received by the client. In addition, four small peaks appear due to the contention of background and video traffic at the physical layer (ethernet).

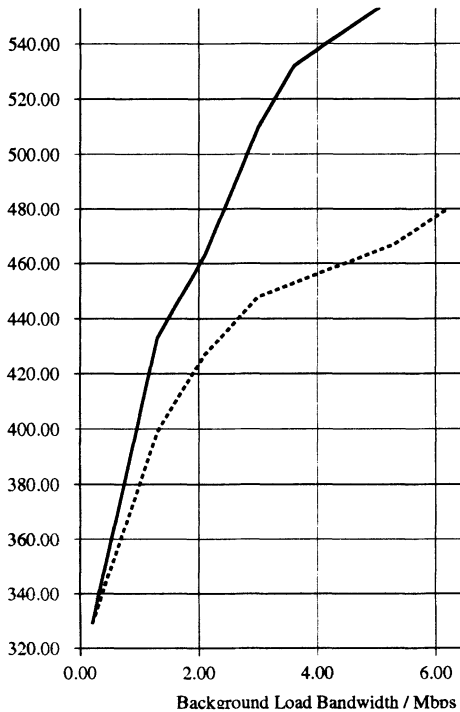
The final testcase in this experiment involved the deployment of the TC (as shown in Figure 2 on page 6) and the launching of a video stream in the presence of the same 6.2 Mbps background network traffic. The results of the measurements are displayed in Figure 5. As can be seen, the main peak has a much higher height than that of testcase 2 (Figure 4). Also the width of the peak has reduced from 0.45ms to 0.3ms - a reduction of 33%. Secondly, the two peaks on the right side of the main peak have been consolidated into a single small peak with minimal width. The TC is unable to remove this peak because it does not have control over the delays that occur as a result of high collision rates between the video and background traffic at the physical layer. On a final note, the small peaks to the left of the main peak have virtually been eliminated. The results of Figure 5 are quite encouraging as they reveal the TC's ability to improve the quality for a video stream in the presence of heavy network traffic.

Figure 6 shows the comparison of the interarrival standard deviations for different levels of network load with and without the TC deployed. As can be seen from Figure 6, the two lines diverge with increasing background load. As expected, the TC appears to be showing more value as the background traffic load increases.

## **Experiment 2 - Behavior of TCP Interactive in the presence of network traffic**

FTP sessions were used to setup TCP connections between Linux 1 and file server (see Figure 2). Without the traffic conditioner, it was seen that transfer of large files consumed average bandwidth of 488 kbps.

The same test was repeated with the Traffic Conditioner deployed. The administrator specified rate for the TCP Guaranteed class of traffic was set to 300 Kbps. The file transfer continued to attempt transferring files as fast as it could. For the first few seconds, the data transfer rate for the flow remained at 488 Kbps.



**FIGURE 6. Standard Deviation of Packet Interarrival Times for Video**

However, the mechanism of the Traffic Conditioner based on queueing delays worked to effect end-to-end TCP rate control.

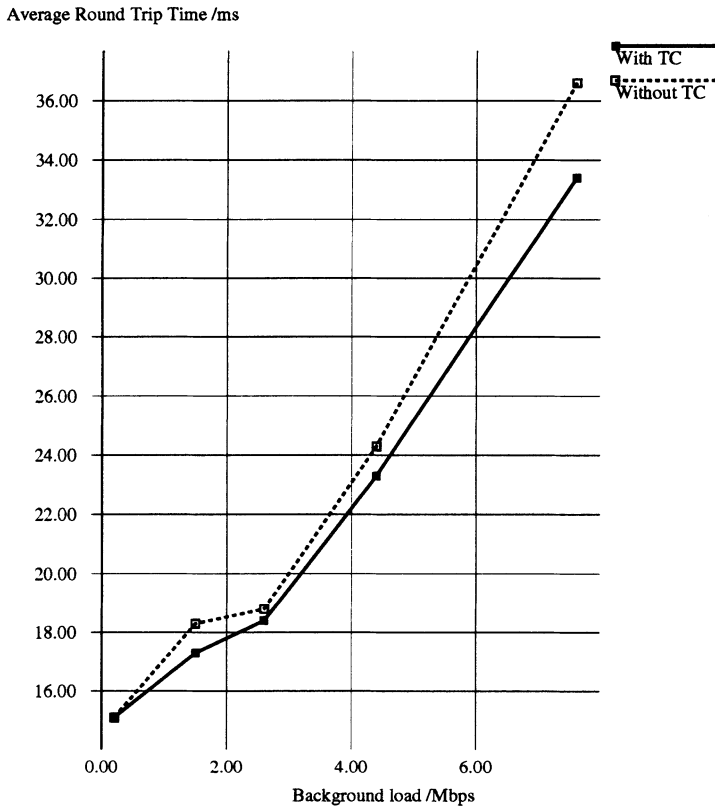
The result being that the application adjusted its sending rate of data to the amount that the Traffic Conditioner would allow through the pipe. The administrator specified rate was 300 Kbps and measurements at the Traffic Conditioner output showed that the flow can be clamped down to 292 Kbps - to an accuracy of 97% percent. The maximum queue length for this class was observed to be 11.

The experiment was also performed with two ftp sessions attempting to transfer large files. It was observed that the two flows shared the available bandwidth of 300 Kbps for this class. One flow was transferred at 130 Kbps and the other at 166 Kbps. The maximum queue length was observed to be 21 for this class of traffic.

### **Experiment 3 - Behavior of TCP Interactive in the presence of network traffic**

The goal of this experiment was to see whether or not the round trip time of TCP Interactive class of traffic suffered in the presence of background traffic. A software tool (HTTP requester) was used to generate HTTP requests to retrieve a file from a web server. The web server runs on the Linux 3 (Figure 2) machine. Video was

played at the background to generate high priority traffic. Also, traffic of UDP Best Effort class is generated as background traffic.



**FIGURE 7. Average Round Trip Time for TCP Interactive Traffic**

The round-trip time for each HTTP request was measured to represent a quantitative indicator of user-experienced delay when using a web browser. The measurement was performed under varying network load. Each measurement is the average of 100 HTTP requests' round trip time. Figure 7 shows the round trip times as a function of network load. It is observed that the RTT is lower when TC is deployed. This is due to the fact that a certain bandwidth is preserved for the Interactive class of traffic. The impact of TC is more prominent at high background load. However, it is debatable if a reduction of 2 milli-seconds of RTT has any impact on the user perception of web browsing.

## 5 CONCLUSION AND FUTURE WORK

The key contribution of this work is to show that it is possible to automatically discover the application QoS requirements without RSVP-like pre-negotiation. Further, experimental results have shown that the on-the-fly classification scheme proposed in [5] can successfully classify the current traffic in IP networks. This implementation has also demonstrated that the combination of classifier, scheduler and admission controller can effectively condition and manage traffic on a link. It has been shown that the different classes of traffic can be serviced in a "required" manner in the presence of background load.

The Traffic Conditioner is one building block in providing QoS capability in IP networks. Investigation and experimentation is needed for other building blocks. The other key building block is a mechanism for providing end-to-end allocation of service without a per-flow connection-setup protocol. This is an area of on-going work.

One open issue is to understand if the flow-based connectionless approach outlined in this paper is scalable. Although this approach removes the scalability issue associated with per-flow state saving at routers, there is still an issue with the number of flows that need to be handled by Traffic Conditioning elements. A second area of exploratory work is to investigate the suitability of the Traffic Conditioner as an edge device [8] in a Differentiated Services network [12].

## 6 ACKNOWLEDGEMENTS

We would like to thank Steve Jaworski for making different software tools available and Joseph Thoo for discussion during various phases of this work.

## 7 REFERENCES

- [1.] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview", June 1994, Request for Comments: 1633
- [2.] J. Wroclawski, "Specification of the Controlled-Load Network Element Service", September 1997, Request For Comments: 2211
- [3.] S. Shenker, C. Partridge, R. Guerin, "Specification of Guaranteed Quality of Service", September 1997, Request for Comments: 2212
- [4.] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", September 1997, Request for Comments: 2205
- [5.] A. Chapman and H. T. Kung, "Automatic Quality of Service in IP Networks", Proceeding of the Canadian Conference on Broadband Research, Ottawa, April 1997, pp. 184-189
- [6.] R. Guerin, S. Herzog and S. Blake, "Aggregating RSVP based QoS Requests", Internet Draft, November 1997

- [7.] K. Nicholas, V. Jacobson and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", Internet Draft, draft-nichols-diff-svc-arch-00.txt, November 1977
- [8.] D. Clark, and J. Wroclawski, "An Approach to Service Allocation in the Internet", Internet Draft, draft-clark-diff-svc-alloc-00.txt, July 1997
- [9.] I. Wakeman, A. Ghosh, J. Crowcroft, V. Jacobson and S. Floyd, "Implementing Real Time Packet Forwarding Policies using Streams", Usenix 1995 Technical Conference, January 1995, New Orleans, Louisiana, pp. 71-82.
- [10.] S. Jamin, P. B. Danzig, S. J. Shenker and L. Zhang, "A Measurement-based Admission Control Algorithm for Integrated Services Packet Network", Proc. of ACM SIGCOMM'95, pages 2-13, 1995.
- [11.] S. Floyd and V. Jacobson, "Link-sharing and Resource management Models for Packet Networks", IEEE/ACM Transactions on Networking, Vol. 3, No. 4, August 1995
- [12.] Nichols, K. and Blake, S., "Differentiated Services Operational Model and Definitions", Internet Draft, draft-nichols-dsopdef-00.txt, February 1998

## 8 BIOGRAPHY

Biswajit Nandy was awarded a PhD degree in the area of multiprocessing in electrical and computer engineering at the University of Waterloo, Canada, in 1993. Since then, he has worked in the area of data communication, in particular, design and development of various building blocks (e.g., cache coherence mechanism, IP-QoS mechanism etc.) for data equipments. His other areas of interests are parallel algorithms, parallel simulation and engineering optimization problems.

Nabil Seddigh, Design Engineer, Computing Technology Labs, Nortel with special focus on next generation IP Networks. He has over five years experience in the telecommunications industry including kernel, file system and device driver design for data communication switches. His recent focus has included the following: IP QoS, Router Accelerator and IP Multicast.

Alan Chapman, Director, Next Generation Systems, Nortel, has over forty years experience in the telecommunications industry and has been a key driver in several of Nortel's major products. For the last ten years he has concentrated on the evolution of data networks. Previous activities include the development of credit based flow control in ATM networks. But more recently, he has been studying the requirements for control of performance in IP networks and the effect on the underlying transport. Mr. Chapman has been awarded many patents in the area of telecommunications systems and next generation networks.

Jamal Hadi Salim, Design Engineer, Computing Technology Labs, Nortel with special focus on next generation IP networks. He has over five years experience in different areas of the telecom industry mainly within Nortel.

# **Part Seven**

---

## **IP/ATM Internetworks**

# An implementation of a gateway for hierarchically encoded video across ATM and IP networks

*Jean-michel Robinet, Yuhang Au, and Anindo Banerjea*

*Electrical and Computer Engineering*

*University of Toronto*

*CANADA M5S 3G4*

*Email: {jrobinet, banerjea, yau}@comm.utoronto.ca*

## **Abstract**

This paper describes an implementation of an application level gateway for connecting adaptive applications using hierarchical encoded video across ATM and IP networks. The gateway participates in RSVP and ATM signaling. The signaling information, as well as local information about processing load, is used by the receivers to decide the number of layers to join, and by the sources to fine tune the bitrates of the layers to the available capacities. Our approach pushes layering related complexity to the edge of the network, and allows us to use standard ATM UNI and RSVP signaling. The gateway participates in a modified session directory (SDR) protocol, to learn the addressing information necessary to perform signaling translation, and to enable layered sessions to be visible across the IP/ATM boundary. By considering all aspects of the problem, especially session directory issues and dynamic bandwidth selection for the layered hierarchy, we have implemented a system that is much more complete than any of the previous prototypes of layered multicasting. This paper describes the implementation experience and presents some measurements of the performance of the gateway.

## **Keywords**

Layered multicast, Internet Protocol (IP), Asynchronous Transfer Mode (ATM), Gateway, IP/ATM interoperation.

## 1 INTRODUCTION

In this paper, we consider multicast video applications that use *hierarchical encoding* to handle network heterogeneity, using signaling protocols to probe the network for available capacity. Multicasting is a powerful network abstraction to support point-to-multipoint and multipoint-to-multipoint communication. Every packet sent to a multicast group is delivered by the network to *all* the receivers of the group. Both Internet Protocol (IP) and Asynchronous Transfer Mode (ATM) networks support multicasting, since it is more efficient than multiple point-to-point connections for the same purpose.

Network environments are heterogeneous by nature. This heterogeneity comes from many sources such as link capacity, end stations processing power and display resolution, network protocols and level of Quality of Service (QoS) support. Heterogeneity is especially problematic for multicast applications, since the receivers may not agree on the data rate that they want to receive, or the protocol to use to signal QoS requirements to the network.

Shacham (Shacham. 1992) has proposed multicast transmission of layered video as a solution to data rate heterogeneity. The data is encoded into a low resolution base stream and a series of enhancement streams. This allows different receivers to receive data from the same source at different rates, simply by subscribing to different numbers of multicast streams, identified by multicast address in IP or multicast virtual circuit (VC) in ATM.

The problem of protocol heterogeneity can be solved by mandating a universal protocol, such as IP. However, this requires unnecessary translations, and prevents applications from taking advantage of the native signaling, when the communication is restricted to a single signaling domain. An alternative approach is to perform translation of signaling and QoS semantics at the network boundaries. This approach allows applications in different networks, such as IP and ATM, to communicate with each other, while continuing to take advantage of the native signaling or resource reservation protocols locally. A possible future network scenario involving this approach is a video on demand system, with the sources being video studios directly connected to a high speed ATM backbone, some high end (perhaps HDTV) clients connected directly to the ATM network, and some lower end clients connected over a slower speed IP-based network. There is no need to involve IP software overheads in the data transmission path from the source to the HDTV clients, but at the same time IP provides access to a more heterogeneous and broader set of clients. This is the model we assume for the rest of the paper.

The paper proceeds as follows: Section 2 presents related work and motivation. Section 3 provides some background on application and session layer issues. Section 4 describes the gateway implementation. Section 5 discusses the implementation, the testbed, and the performance of the gateway. We conclude the paper with a summary in Section 6.



## 2 RELATED WORK AND MOTIVATION

This paper describes an implementation of a gateway to allow adaptive layered video applications to communicate across different protocol domains, specifically IP and ATM networks. Our work is related to prior work on Heterogeneous MultiCast (HMC) (Sudan *et al.* 1997) with a IP/ATM gateway for layered video, but differs in certain key areas that we describe below. In addition, the applications considered in HMC are nonadaptive. Our work is also similar in many respects to Receiver-driven Layered Multicast (RLM) (McCanne *et al.* 1996), which uses receiver adaptation based on packet loss to select the number of layers to receive, but is restricted to an all IP environment. Thus, many of the new problems we face are related to the handling of adaptive applications in a multiprotocol signaling environment.

Both of the above systems only allow the receiver to select from a static set of layers, based on network bandwidth availability. If the set of bandwidths being transmitted is not well tuned to the set of bandwidths available, these systems perform poorly. We believe that it is unreasonable to require the user to know *a priori* the set of network and receiver capacities required, and configure the sources with the correct set of layer bitrates. We implement feedback mechanisms across the network, so that the bitrates transmitted on the layers from the sources are adapted to the set of network bandwidths and receiver capacities dynamically. This functionality is orthogonal to the functionality provided by SCUBA (Amir *et al.* 1997), where the information from different sources is dynamically mapped on to a static set of layers in response to receivers expressing interest in particular sources. Our adaptation model also allows us to push all layering related complexity to the application layer, using standard User to Network Interface (UNI) (UNI 3.0. 1993) signaling within the ATM network and ReSerVation Protocol (RSVP) (Zhang *et al.* 1993, Braden *et al.* 1996) signaling within the IP network, instead of modifying the protocols to handle layering as suggested in HMC.

Previous work in layered multicast has neglected the problem of session advertisement. The session directory information provided by SDR (Handley *et al.* 1995) allows a receiver to learn of the existence of a session, and provides sufficient information (such as multicast addresses and port numbers in IP) so that the receiver can join the session. In an IP/ATM layered environment, the problems of how layered sessions are specified in session advertisement messages, how receivers in one domain learn about sources in the other domain, or how to reconcile the multipoint-to-multipoint nature of IP multicast with the point-to-multipoint nature of ATM virtual circuits (VCs), have not been previously dealt with. For example, in Sudan's work the gateway must be manually configured with a *static* mapping between layers, IP multicast addresses, ATM addresses, participant identifiers (independent of domain specific addresses), and traffic descriptors for the layers (in the ATM to IP case).

Our work addresses these problems in a more general way, by extending

the Session Directory (SDR) protocol to handle layered sessions, ATM addresses, and the point-to-multipoint nature of ATM multicast. The gateway participates in the session directory protocol in both domains and performs session advertisement translation. These issues are particularly important for our system, since the applications are adaptive and use the session directory protocol to advertise *changes* in the structure of layers transmitted by the sources, in response to feedback about network and receiver heterogeneity.

The approach of RLM is targeted to an all IP environment, and does not depend on the existence of signaling protocols (e.g., RSVP). If RSVP signaling is available, receiver adaptation can be much more stable than in RLM, while at the same time responding very quickly to available capacity. In RLM, stability and response time must be traded off against each other, depending on complex interactions between the duration of congestion and the time the network takes to prune a connection after a receiver leaves. Since prune times of currently deployed multicast routing and group membership protocols in the Internet are long (Gupta *et al.* 1997), a method like RLM must necessarily have poor response times to be stable (McCanne *et al.* 1997). In addition, using RSVP and ATM UNI signaling allows us to provide QoS guarantees.

Finally, Sudan *et al.* do not consider the case of connecting an ATM network to an IP network without RSVP support. We handle the case where RSVP support is not available, by using a loss based mechanism adaptation similar to RLM. We help to handle some of the stability and latency problems raised by this approach, by performing priority service based on layer number at the gateway, to concentrate loss due to congestion on the highest layers. This provides a clearer signal to the receiver to adapt, while reducing the impact of the loss on the received video quality.

Our applications are based on Zakhor's software codec (Taubman *et al.* 1994), which is capable of encoding digital video into a very large number of fixed size sublayers. Previous work (Banerjee *et al.* 1997) showed how these can be combined dynamically to a smaller number of transport layers, allowing the bitrate to transport layer mapping to be adapted by the source in response to network feedback. We have added signaling support (RSVP over IP and UNI over ATM) and adaptation mechanisms to the above, to create adaptive layered network conferencing and video server applications.

### 3 APPLICATION AND SESSION LAYER PROCEDURES

In this section, we briefly summarize the feedback algorithms used by the receiver and the source to adapt to the network capacity and receiver load, and the signaling and session directory functionality required to handle layered applications on IP and ATM. Details can be found in (Yau *et al.* 1997).

Our layered application uses three different adaptation mechanisms, which work over different time scales and distances. The first is adaptation to receiver load. Since our application performs decoding of the layered video in software,

the number of layers it can receive depends on the CPU load. The receiver monitors the time to process a frame of video to determine the number of layers it is able to process. This feedback is entirely local to the receiver's machine and involves the shortest time interval.

The application uses network signaling mechanisms (RSVP over IP and UNI over ATM) to determine network bandwidth availability. When the receiver load allows, the receiver probes the network with a reservation attempt to add the next layer. There is no danger of unstable behavior, since the layer is only added if its addition cannot cause congestion. This feedback involves the network and occurs over slightly longer intervals than CPU Monitoring.

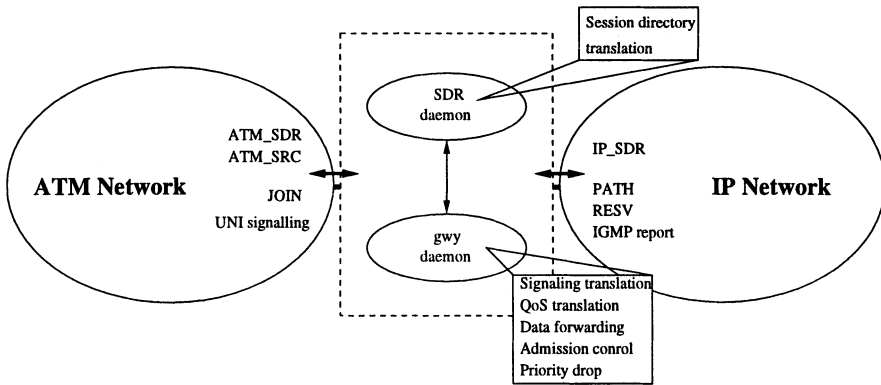
In the absence of signaling mechanisms (for example, IP networks without RSVP) the application uses a loss based feedback mechanism similar to RLM for network adaptation. However, this leaves us with the problems of stability and responsiveness mentioned before. We return to this issue in Section 4.

The receivers provide feedback to the session originator about the link capacities and receiver processing powers of the active receiver and network environment. The originator adjusts the bitrates being transmitted on each layer of the encoding hierarchy accordingly, and advertises the changed layer hierarchy information to the sources using the session directory protocol. The sources modify their transmitted hierarchies accordingly. This feedback involves both sources and receivers. For scalability reasons, it occurs over the longest time interval. Note, however, that this does not imply the system is not responsive, since the receiver adaptation (compared to RLM) is fast. The system adapts slowly to changes in the receiver set, and in comparison, all previous approaches did not adapt in this respect at all.

In the IP network, a receiver changes its video quality by joining or leaving multicast groups using the Internet Group Management Protocol (IGMP). An IP multicast group is a many-to-many communication abstraction, so the receiver receives data from all sources transmitting to it. The receiver must also use RSVP to specify its QoS requirements, and make reservations for the layer. By waiting till the RSVP reservation request is successful before adding the next layer, the receiver can ensure that the network is not overloaded.

In the ATM domain, a receiver changes its fidelity by joining or leaving a multicast Virtual Circuit (VC). In ATM 3.x, the join request can only be initiated by the source; hence, the receiver must send a request to the source to be added to a specific layer. An ATM multicast VC is strictly one-to-many, so the ATM receiver must know the Service Access Point (SAP) address of each source in order to send the requests to join. The Session Directory (SDR) protocol must be modified to carry this source specific information. We accomplish this by adding a ATM\_SRC message, which is transmitted by each source on the ATM network, and carries the address information.

We also extend the SDR message to carry layering information. This message is periodically retransmitted by the session originator, and conveys information about the number of layers, and the multicast address, and bitrate



**Figure 1** Gateway design

associated with each layer. Based on feedback received from the receivers, the originator transmits a changed SDR message. On receiving the new SDR message, sources adapt their transmitted streams; receivers can detect and adapt to the change in the data stream.

## 4 GATEWAY PROCEDURES

To maintain transparent interdomain connectivity, the gateway performs the following tasks:

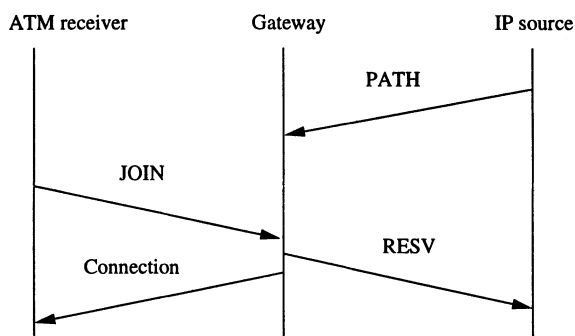
- Translation of the connection setup messages.
- Translation of the traffic parameters.
- Translation of the session directory messages.
- Admission control.
- Data forwarding.
- Priority service.

As shown in Figure 1, two daemons, the SDR daemon and the gateway daemon, are responsible for the above tasks. The SDR daemon is only responsible for translating session advertisement messages from one domain to the other one. The gateway daemon is responsible for the other tasks.

### 4.1 Signaling translation

The gateway has to translate the signaling messages and map the traffic parameters between the two domains. In the following subsection, we consider the case of an IP source and an ATM receiver, and after that, the case of an ATM source and an IP receiver.

*IP to ATM:* The gateway learns of the existence of a new session from the session directory (SDR) protocol. It joins the multicast groups in order to



**Figure 2** End-to-end signaling: IP to ATM case

receive PATH messages for each group and hence learn the existence of an IP source. The gateway also uses SDR to announce the session on the ATM network. Figure 2 shows how the gateway translates the messages to have a coherent end-to-end signaling.

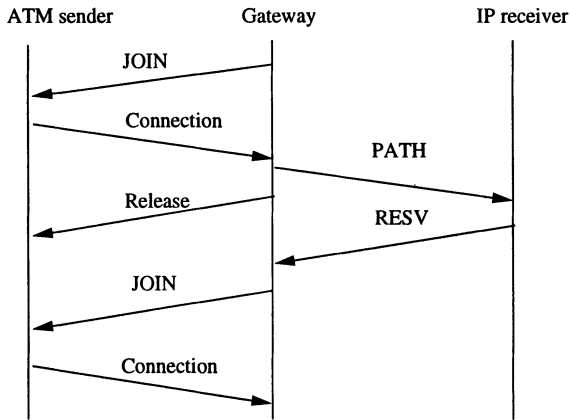
The gateway learns of a new source when it receives a new PATH message on the base layer multicast address. After performing local admission control tests, the gateway behaves like a new source on the ATM network, and advertises its address in an ATM\_SRC message. It also adds the layer by creating a multicast VC (with currently no receivers) for this layer with the QoS translated from the PATH message, and updating the database.

When an ATM receiver wants to receive a particular source and layer, it sends a JOIN request to the gateway. On receiving such a JOIN, the gateway looks up the corresponding IP source from the database, and makes a reservation, using a RESV message to the IP source. Finally, the gateway adds the receiver to its ATM multicast VC, and updates the forwarding table.

In case RSVP signaling is not present, the gateway will not be able to depend on PATH messages and NULL PATH messages to learn of the joining and leaving of IP sources. In this case, it joins all multicast groups, and uses the presence of data to learn about sources. On learning of new sources, it advertises them using ATM\_SRC messages. It uses timers to learn about sources leaving. In order to assist the receiver in performing network adaptation, the gateway performs priority service on the layers of the same session. This is further described in subsection 4.3.

*ATM to IP:* When a session is created by an originator on the ATM network, the gateway learns about it from an SDR message. The gateway SDR daemon acts as an originator on the IP side, by choosing IP multicast addresses for the session and sending SDR messages. Figure 3 shows how the gateway translates the signaling messages, when the source is in the ATM network and the receiver is in the IP network.

The gateway learns the existence of a new ATM source by receiving an ATM\_SRC message. The gateway sends as many JOIN messages as layers to the ATM source. It translates the QoS learned from the ATM signaling, performs the local admission control tests, and sends as many PATH messages



**Figure 3** End-to-end signaling: ATM to IP case

as existing layers to the corresponding multicast groups. The gateway then tears down the multicast VCs, since no receivers exists yet. For each ATM source, the gateway is seen as a different IP source, distinguished by source UDP port number, by the receivers in the IP domain.

The gateway learns about the first IP receiver for a layer by receiving a RESV message on a multicast group. After performing local admission control, the gateway sends a JOIN message to the ATM source requesting the particular layer. For subsequent IP receivers, no new state needs to be setup at the gateway, and the RESV message is not even forwarded to the gateway process by RSVP unless the reservations change.

If the gateway does not support RSVP, the signaling is very simple. The gateway learns the existence of a new ATM source by receiving an ATM\_SRC message. The gateway waits for an IP receiver to show interest by sending an IGMP report message. Once a receiver exists, the gateway sends a JOIN message to the ATM source, updates the database and forwarding table, and forwards the data to the IP side. The gateway also learns about receivers leaving using from the IGMP protocol, and deletes the corresponding ATM connections when all receivers for a given multicast address leave.

## 4.2 Data forwarding

To forward the data streams from the senders to the correct set of receivers, the gateway must identify: (1) the sender, (2) the layer, and (3) the receivers.

When a packet is received on the IP side, packet header contains the IP multicast address, the UDP destination port, the IP source address, and the UDP source port. Based on this four fields, the gateway looks up a multicast VC handle from the forwarding table and transmits the data. When a packet is received on the ATM side, the gateway uses the multicast VC handle to look up the addresses to put on the outgoing packet header in the forwarding table.

Note that the forwarding state of the gateway is extremely simple. Packets are actually queued in the receive socket buffers (in the kernel). The gateway daemon processes packets one at a time, reading from one socket and immediately transmitting to another. In order to implement priority service, it just associates a priority with each socket, and serves them in priority order. In order to implement controlled discarding (such as RED In Out (RIO)), it uses IOCTL calls to read the socket buffer length. This results in a very simple and robust gateway design.

### 4.3 Participation in adaptation procedures

Where the gateway acts as a receiver into a network, either ATM or IP, it ensures that it adds and drops layers in order of the layer number. This functionality is redundant, since the receivers exhibit this property individually. However, it helps to stabilize the system against the effect of accidental use of the multicast address space of a layered session by an unrelated receiver, or against incorrect behavior on the part of layered receivers.

The gateway is responsible for forwarding feedback messages from the receivers to the originator of the session across the ATM/IP boundary. This allows the originator to get the full picture about the set of receiver and network capacities, so it can make its decision about changing the transmitted hierarchy.

When the originator decides to change the layer hierarchy, it transmits a new SDR message. These messages are translated and forwarded from one network to the other by the gateway. This allows sources on both sides of the network to learn about the new hierarchy. The sources then start to transmit data according to the new hierarchy. On the IP network, they start sending the data according to the new scheme right away, and also transmit PATH messages to notify the network and the receivers about the changed resources needed. If resources are available, the reservations are modified by RESV messages from the receivers without a need to tear down the current distribution tree. On the ATM network, the sources tear down the current distribution tree and wait for new requests to join from the receivers. The new multicast VCs are set up according to the modified bitrate requirements.

The gateway participates in this adjustment of the distribution trees. The signaling actions taken by the gateway as part of the adaptation process are all the result of messages from the network sent by the receivers (e.g., RSVP RESV messages or ATM JOIN request), or the senders (e.g., RSVP PATH messages or ATM\_SRC messages). The gateway never initiates any adaptive action, therefore its state is simple. For example, no timers need to be kept.

However, in the absence of RSVP, the gateway must detect the coming and going of IP sources by using timers, since no explicit notification is provided by the network. The gateway also assists the adaptation process at the receiver by performing priority service of packets, based on layer number. The receiver performs loss based load shedding similar to RLM in the absence of RSVP.

Priority service concentrates the loss onto the highest layers in the case of congestion. This has two advantages. Firstly, the receiver can compute the loss rate for each layer separately, and this will be much higher than if the same loss was spread across all the layers. This gives a clearer signal to the receiver to drop the highest layer. For instance, the drop trigger thresholds can be set higher, so a few accidental packet drops do not cause the receiver to back off. This makes the adaptation more stable while still remaining responsive. The second advantage is that if the loss is concentrated on the highest layer, the visual quality of the video is less effected than if the same loss is spread across all the layers.

## 5 IMPLEMENTATION STATUS AND PERFORMANCE

The current implementation of the gateway at the user level has several limitations. The number of sessions that can be simultaneously handled is severely limited by the number of sockets available to a single UNIX process. The gateway has also not been optimized for high throughput. Finally, the support for local admission control of the gateway resources and controlled dropping of low-priority packets under overload is not complete. The version of the gateway that we used for the measurements implements simple priority service, with the priority based on layer number and QoS support.

The gateway runs on top of the socket interface, and binds virtual circuits on the ATM network and multicast groups on the IP network to sockets. Thus, for example, for a single session with seven layers, one ATM sender, one IP sender, and arbitrary number of ATM and IP receivers, the gateway daemon uses twenty three sockets. Since a UNIX process has access to a maximum of 64 sockets, this limits rather severely the size and number of the sessions we can create. One possible solution to this problem lies in moving the gateway implementation into the kernel. This would reduce the memory copy overhead as well, leading to an improvement in the maximum throughput capacity. This may be appropriate for a stand alone machine, with the sole function of connecting layered applications across ATM and IP.

The following experiments were conducted on a network testbed consisting of an ATM LAN and a 10 Base-T Ethernet LAN. The ATM LAN consists of a Fore ASX-200WG switch, with multiple UltraSparc workstations connected using Fore SBA200 ATM interface cards, and running ForeThought 4.1.0 driver software that implements an application programming interface to the ATM UNI 3.0. The machines on the Ethernet LAN run the ISI implementation of RSVP on Solaris 2.5.1. The ATM and Ethernet LANs are connected by a Sparc20 workstation running Solaris 2.5.1 with ATM and Ethernet network interface cards. This workstation runs the gateway software.

Our implementation is not optimized for fast forwarding performance. For example, it would be possible to move the forwarding function into the kernel to lower memory copy overheads and improve throughput. Hence, the first



experiment we performed is to test the maximum data forwarding capacity of the gateway. We found that even with multiple sessions running simultaneously, the gateway is capable of forwarding data up to the full capacity of the Ethernet without overload. The signaling performance is also not degraded at this high load, indicating that the gateway has sufficient CPU cycles to spare.

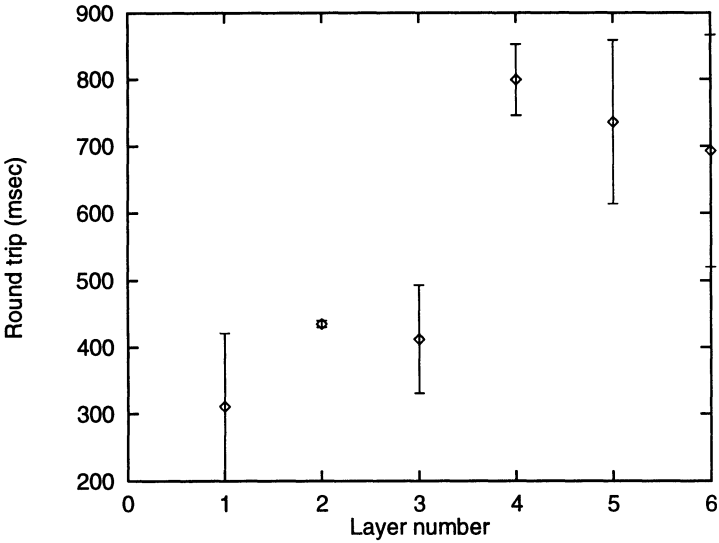
The next set of experiments presented explore the time to add a layer. Our receivers use signaling to probe the network for available capacity. In the worst case, this requires one round trip for each layer; the latency of this process is of concern.

Figure 4 shows the round trip latency to add a layer with a single source on the ATM network, and a single receiver on the IP network. The round trip is measured from when the RESV message is transmitted by the receiver, to when the first data packet arrives at the receiver. It is important to note that the case of a single receiver is the worst, since it requires a full round trip for each layer. A second receiver on the Ethernet would observe much less latency; it would start receiving the data as soon as it bound a socket to the multicast address, as the data is already being transmitted on the LAN.

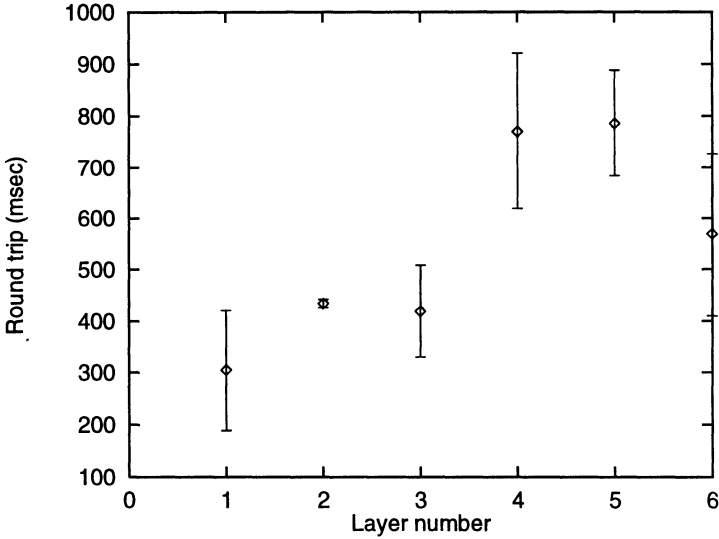
The X-axis shows the layer number  $N$ , while the Y-axis shows the time to add the  $N$ th layer averaged over eighty repetitions of the experiment, and the 95% confidence intervals. We note that the average time to join a layer increases with the number of layers being added. Figure 5 shows the IP to ATM case. Table 5 shows the breakup of the round trip time into its major components, shown as a (MAX, AVERAGE, MIN) triple in units of milliseconds. These components are:

- **RSVP processing:** the delay in the receiver host from when the decoder decides to add a layer to when the RESV message is sent. This processing is performed on the receiver, by the RSVP library and daemon code. The processing time of layer 1 is greatest, because an RSVP thread is created.
- **RESV processing:** the delay in the gateway machine from when the gateway receives a RESV message to the time the multicast VC is setup. This involves setting up the internal tables, sending a JOIN message to the ATM source, and then performing ATM UNI signaling.
- **Data sent:** the delay from the time the VC is setup to the time the first packet arrives at the gateway. A major component of this delay is the time spent waiting for the next round of transmission. In addition, for layer 1 the source has to wake up a thread before data is transmitted.
- **Forwarding:** the delay in the gateway from when the data is read to when the data is transmitted. This is not shown in the table since it is almost fixed at 1 ms.
- **Data read:** the delay at the receiver, from when the data arrives in the kernel buffer to when the application reads it from the kernel buffer.

We note that the major components of the delay, as well as of the variability, are the time to send the data from the source and the time to receive the data



**Figure 4** Round trip latency for adding a layer: ATM to IP

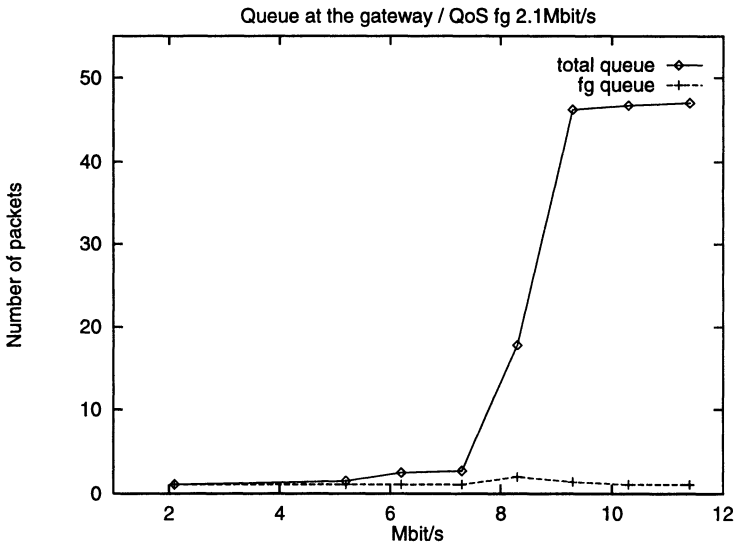


**Figure 5** Round trip latency for adding a layer: IP to ATM

at the receiver. At the source, this time is the time from when the ATM connection has been successfully set up to when the next frame of data is due to be transmitted. Since the encoder places 4 frames of data into a single packet and the frame rate for this experiment was ten frames per second, the inter-transmission time is 0.4 seconds. We see that the send time is spread between 0 and 0.4 seconds as expected. At the receiver end, the time to receive the packet is also similarly affected by the time to process a frame,

**Table 1** Break up of time to add a layer; ATM to IP case

<i>Layer</i>	<i>RSVP proc.</i>	<i>RESV proc.</i>	<i>data sent</i>	<i>data read</i>
0	(18/23/81)	(37/41/64)	(25/232/409)	(0.3/0.4/0.7)
1	(3/6/10)	(36/43/140)	(170/373/382)	(0.2/0.4/0.7)
2	(3/5/11)	(36/43/173)	(101/135/484)	(210/234/290)
3	(3/5/7)	(124/149/217)	(308/373/402)	(230/271/332)
4	(3/5/7)	(55/115/196)	(286/375/418)	(2/260/312)
5	(3/5/15)	(35/72/123)	(238/358/385)	(2/317/331)

**Figure 6** Gateway queue size

since while the decoder is processing the previous frame it does not check if new data has arrived. When the decoding threads become idle, the thread which is blocked on the receive gets a chance to run and retrieve the data from the network. This time increases with layer number and saturates near 0.4, since the receiver does not take on more layers than it can handle. Thus, the major element affecting the time to add a layer is the unit of transmission, which in this case happens to be four video frames.

In order to maintain stability, the receiver has to perform some measurements of the last layer added before it can safely add the next layer. The intervals of time for this measurement increase as the receiver becomes loaded, to avoid oscillatory behavior. These intervals are of the order of seconds, which is quite large compared to the round trip time to add a layer. Thus, the signaling latencies are acceptable for the applications under study, and optimizing the signaling overhead by using 'bundled VCs' as proposed in HMC is not necessary.

Figure 6 shows the queue lengths for two simultaneous streams. The foreground stream is a 2.1 Mbps stream, carried with QoS support, using ATM and RSVP signaling. The background stream is a best effort stream with bitrate increasing from zero to nine megabits per second. The X-axis shows the total bitrate of the combined traffic, while the Y-axis shows the queue length for foreground queue (diamond symbols) and total queue length (plus symbols). We note that the foreground queue remains small, even when the sum of bitrates exceed the capacity of the Ethernet (10 Mbps). This happens because of two reasons. Firstly, since the foreground traffic is shaped at the entrance to the ATM network, a burst of traffic is never injected in this stream. Secondly, the QoS stream is protected from the bursty behavior of the best effort background stream by the per stream queueing and priority service in the gateway (as well as in the switches, etc). This experiment shows the effectiveness of the priority service and QoS mechanisms; even under overload conditions, all the queueing delay and loss are concentrated on the low priority stream.

## 6 CONCLUSION

We have presented an implementation of a gateway for connecting layered applications across ATM and IP networks. This implementation improves on previous research by extending the feedback algorithms for adaptation all the back to the source. This allows the source to select the correct number of layers, and the bitrates for each layer, to accommodate the current network and receiver capacities. Our adaptation model has three different control loops, one limited to the receiver, a longer one involving the receiver and the network, and a third (longest) control loop involving the source, network and receiver. The combination of the three gives us allows the feedback to be scalable, stable, and still responsive. The gateway participates in the source adaptation by translating the feedback messages, and updating the layered hierarchy advertisements (in the session directory protocol) when they change.

We also extend previous research by considering the addressing and naming translation issue. The extension of the session directory protocol to the ATM environment allows us to compensate for the lack of a multipoint-to-multipoint abstraction on the ATM network, since the receivers can find out about source information from the session directory. The gateway participates in the session directory protocol, to become aware of new sessions and sources, to advertise them on the other side, and to translate the addresses, port numbers and other information that the receivers need to join a session. The gateway acts as a proxy for each IP source on the ATM network, and acts on behalf of IP receivers on the ATM side. Instead of using preconfigured tables for the address translation, the gateway exchanges the necessary information through the session directory protocol.

In another departure from previous work, our applications take care of the complexity of layering, such as ensuring that resources are not wasted for

higher layers when lower layers are not available, at the edge of the network. This simplifies the network from the point of view of network scalability. It also allows us to perform our experiments with a standard ATM and RSVP installation. Our experiments show that the signaling is not a major factor in the latency of the receiver based adaptive control.

We deal with the case when RSVP is not present, by using a loss based mechanism similar to RLM. In this case, the receiver responds to congestion by detecting increased packet loss and dropping layers. Since the gateway is itself likely to be a bottleneck (going from a high speed ATM to a low speed Ethernet network), we concentrate loss at the gateway on to the highest layers by performing priority service. This gives the clearest feedback to the receivers, since the percentage of lost packets on the highest layer is maximized. It also minimizes the effect of the loss on the visual quality of the video. Finally, this action, being taken by an application level entity, does not cause any increase in the network complexity or a violation of layering.

In conclusion, we consider all aspects of the layered multicast problem at the ATM/IP gateway. We contend that this makes our system more usable, more complete, more flexible, and more stable than previous prototypes of layered multicasting.

## REFERENCES

- Amir, E. McCanne, S. and Katz, R. (1997) Receiver-driven Bandwidth Adaptation for Light-weight Sessions, *ACM Multimedia*, pp. 415-426, Seattle, WA.
- Banerjee, A. Tan, W-T. and Zakhor, A. (1997) Evaluation of a Layered 3-D Subband Compression Scheme for Multicasting Cine-Angiograms across Heterogeneous Networks, *Proc. SPIE Medical Imaging'97*, pp. 265-276, Newport Beach, CA.
- Braden, B. Zhang, L. Berson, S. Herzog, S. and Jamin S. (1996) Resource Reservation Protocol (RSVP) – Version 1 Functional Specification, *Internet Draft*, Work In Progress.
- Gupta, A. and Speer, M. (1997) Measuring the performance of IP multicasting, *Sun Microsystems Laboratories Document*, *SMLI-97-0048*, Palo Alto, CA.
- Handley, M. and Jacobson, V. (1995) SDP: Session Description Protocol, *Internet Draft*, Work In Progress.
- McCanne, S. Jacobson, V. and Vetterli, M. (1996) Receiver-driven Layered Multicast, *Proceedings of SIGCOMM'96*, pp. 117-130, Stanford, CA.
- McCanne, S. Vetterli, M. and Jacobson, V. (1997) Low-complexity Video Coding for Receiver-driven Layered Multicast, *IEEE Journal on Selected Areas in Communications* 16, 6, pp. 983-1001.
- Shacham, N. (1992) Multipoint Communication by Hierarchically Encoded Data, *Proceedings of IEEE INFOCOM'92*, pp. 2107-2114, Firenze, Italy.

- Shenker, S. and Breslau, L. (1995) Two issues in Reservation Establishment, *Proceedings of SIGCOMM'95*, pp. 14-26, Cambridge, MA.
- Shenker, S. Partridge, S. and Guerin, R. (1997) Specification of Guaranteed Quality of Service, *Internet Draft*, Work In Progress.
- Shenker, S. and Wroclawski, J. (1997) General Characterization Parameters for Integrated Service Network Elements, *Internet Draft*, Work In Progress.
- Sudan, M. and Shacham, N. (1997) Gateway Based Approach For Managing Multimedia Sessions over Heterogeneous Domains, *Proceedings of IEEE INFOCOM'97*, Kobe, Japan.
- Taubman, D. and Zakhor, A. (1994) Multirate 3-D Subband Coding of Video, *IEEE Transactions on Image Processing* 3, 5, pp. 572-588.
- ATM User Network Interface Specification, Version 3.0. (1993) *Prentice Hall*.
- Wroclawski, J. (1996) Specification of the Controlled-Load Network Element Service, *Internet Draft*, Work In Progress.
- Yau, Y. Robinet, J-M. and Banerjea, A. Session and application layer issues for layered multicasting, *document in preparation*.
- Zhang, L. and Deering, S. Estrin D. Shenker, S. and Zappala D. (1993) RSVP: A New Resource ReSerVation Protocol, *IEEE Communications Magazine* 31, 9, pp. 8-18.

## 7 BIOGRAPHY

Jean-michel Robinet received a B.S. in Computer Engineering from the Florida Institute of Technology in August 1996. He is currently a M.A.Sc. candidate at the department of Electrical and Computer Engineering at the University of Toronto. His graduate research is on multimedia networking in heterogeneous environments.

Yuhang Au received his B.S. in Computer Engineering from the University of Kansas in August, 1996. He is currently a M.A.Sc. candidate at the Department of Electrical and Computer Engineering at the University of Toronto. His graduate research is on video conferencing in heterogeneous networked environments.

Anindo Banerjea received a B. Tech degree in computer science and engineering from the Indian Institute of Technology, Delhi in 1989, and a Ph. D. in Computer Science from the University of California at Berkeley in 1994. His dissertation topic concerned the problem of providing fault recovery in networks with guaranteed performance (real-time) services. He was also involved in the design and implementation of the Tenet realtime protocol suite, which provides real-time services on wide area inter-networks. As Assistant Professor at the Department of Electrical and Computer Engineering at the University of Toronto, Anindo is continuing his research in multimedia networking. He is interested in heterogeneous networking environments, integration of Asynchronous Transfer Mode (ATM) technology with the Internet, QoS sensitive multicast routing and real-time networked multimedia applications.

# Trading off network utilisation and delays by performing shaping on VC ATM connections carrying LAN traffic.

*P. Castelli   L. Guida   M. Molina*

*CSELT (Centro Studi E Laboratori Telecomunicazioni)*

*V. Reiss Romoli, 274 – 10148 Torino, Italy*

*Phone: +39.11.228.8784 – Fax: +39.11.228.8862*

*e-mail: {paolo.castelli,luisa.guida,maurizio.molina}@cselt.it*

## **Abstract**

This paper considers a scenario where the traffic of several LANs is transported on Deterministic Bit Rate (DBR) or Statistical Bit Rate (SBR) ATM Virtual Channel Connections (VCCs), that are then multiplexed into a DBR Virtual Path Connection (VPC) with fixed, dedicated bandwidth. It is investigated whether or not it is suitable to shape VCCs according to a DBR or SBR traffic contract before multiplexing. Results show that DBR shaping is rather useless, as with respect to the unshaped case no significant utilisation gain can be achieved without introducing high delays in the shapers' buffers, and that SBR shaping behaves no better, due to the impossibility of finding a typical burst duration and mapping it on SBR traffic descriptors.

## **Keywords**

Shaping, Self Similarity, Long Range Dependence, Multiplexing, IP over ATM

## 1 INTRODUCTION

When talking about ATM technology, its ability to differentiate the Quality of Service (QoS) of the carried traffic streams according to their needs is often mentioned as a graceful characteristic. Real time applications requiring stringent end to end information transfer delays and delay variations can be carried with the higher QoS class, which in the ITU-T terminology is called “QoS class 1” (ITU-T I.356, 1996). Applications more tolerant to delays, like data applications, can be carried either with QoS class 2 or QoS class U. QoS class 2 means that Cell Loss Ratio (CLR) is guaranteed to be lower than a certain bound, whereas QoS class U means that no guarantee at all is given, neither on losses nor on delays. Although data applications always have the ability to detect and recover packet losses through frame retransmission, a lot of simulation studies have shown how dramatic can be, in terms of increased end to end delays, the effect of unbounded cell losses during congestion periods (Bonaventure, 1997). In spite of being in principle “tolerant” to application delays, users of applications like Telnet, FTP, Web Browsing, etc., would greatly appreciate the performance improvement coming from limited cell losses in their data.

Right after the ability of ATM to differentiate traffic into QoS classes, it can be recalled its scalability, i.e. its suitability to be used both as a LAN and as a WAN technology. In recent years, a lot of ATM LANs have been deployed and they are nowadays being used with success. Also, trials and experiments to deploy and operate ATM in the WAN environment have been performed, and ATM backbone networks are now a reality.

In parallel, especially due to the booming growth of the Internet, IP protocol has consolidated its positions, and legacy LAN technologies like the Ethernet have been significantly improved (100 Mbit/s Ethernet switches being already widely deployed and Gigabit Ethernet being right round the corner).

In summary, ATM can be successfully used as a backbone technology also to carry data traffic relaying on the IP protocol generated on non ATM LANs. The bursty nature of this kind of traffic gives the public carriers the opportunity to achieve some statistical gain (or “multiplexing gain”), but the need to do that while meeting some QoS contract rises a lot of traffic engineering issues, and this paper addresses some of them.

The paper is organised as follows: in section 2 we describe how we performed some traffic measurements over CSELT’s LANs in order to verify some characteristics of LAN traffic (Self Similarity) that had already been described in literature (Leland, 1994) and to obtain the values needed to parameterise the source traffic models we used in the study. Such models are described in section 3. In section 4 we describe the simulation scenario we implemented, while in section 5 we focus on the effectiveness of the traffic shaping as a way to meet QoS commitments. Finally, in section 6 we present some conclusion and outline the future work.



## 2 TRAFFIC CHARACTERISATION

A thorough characterisation of the traffic generated by IP applications over today's LANs has a great importance when performing internetworking studies. After the pioneering work at Bellcore (Leland, 1991), which put into evidence the Self Similar and Long Range Dependence (LRD) characteristics of this type of traffic, a lot of efforts all over the world was made to faithfully reproduce these characteristics by means of models more complicated than the traditional markov ones. A good review of these traffic characteristics and proposed models can be found in (Morin, 1996).

Usually, the path followed by researchers in this area is to perform some measurement on real networks, verify Self Similar characteristics, provide an estimation of the Hurst parameter and of other parameters of interest and then use those values into some traffic models to show how good they are in reproducing the statistical characteristics of real traffic traces and/or their queuing behaviour.

This is what we did in our study too, but we were less concerned about comparing performances of "advanced models". This not because we believe we used the best possible traffic model, but because we are aware that whatever thorough the characterisation of some measured traffic is, it is probably very closely related to the network technology and to the applications and protocols used over it. This danger is very well explained in an article by Paxson and Floyd (Paxson, 1997), which also suggests the wide variation of parameter values used in the models as a method to extend the generality of internetworking studies performed. This was the approach followed in this study.

For the sake of completeness, however, we briefly describe how we collected and analysed measurements on some CSELT's LANs.

We focused on a 10Mbit/s Ethernet segment collecting traffic from several hosts (mainly PCs with windows 95 and Unix workstations, mostly running ordinary network applications such as e-mail clients, FTP and Web Clients, telnet, Xwin, Sun NFS). Measurements were collected by a Sun Ultra 1 Workstation with a 200 MHz processor, with the aid of a modification of the freeware "tcpdump". The main modification consisted in the fact that no single packet information was stored on the disk but only, at fixed time intervals, the summary information about the number of Ethernet bytes seen on the segment. The time interval duration was one second, and measurements were repeatedly collected from 9.00 am to 17.00 pm, for sixteen working days. Some comparison of our data with data collected in parallel with the aid of a Wandell & Goltermann Da-30 protocol analyser, showed that packet losses by the tcpdump modification were limited, and estimated statistical parameters were not significantly affected.

Unfortunately, the 10 Mbit/s Ethernet segment under measurement only collected a limited amount (say, less than  $\frac{1}{4}$ ) of the Intranet and Internet traffic generated/received by the 200 researchers hosted in our building. As a result, the sixteen daily collected traffic profiles often showed some evident nonstationarity. In order to increase stationarity, we superposed them four by four, thus obtaining

four aggregate profiles, being more stationary and potentially more representative of the traffic generated/received by all the researchers of our building. Of course such an operation was possible only because we performed load measurements, and we didn't compute any packet interarrival time statistic.

The four "aggregated" profiles were then analysed in order to compute some statistical parameters. Among them, the more relevant to this study were

- the mean rate (in bytes/s);
- the peak factor, i.e. the ratio of the variance of the number of bytes to the mean of the number of bytes seen at each one second time interval;
- an estimation of the Hurst parameter based on the Index of Dispersion for Counts (IDC). See (Gusella, 1990).

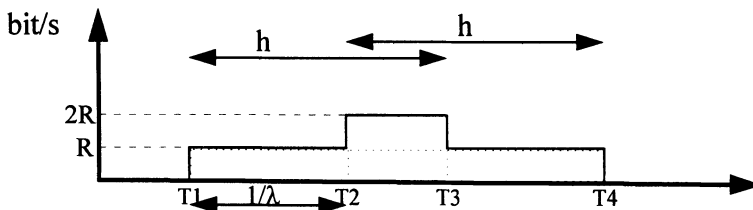
We then chose one of the profiles as being the more representative one, and used the computed values as a starting point to parameterise our model, whose description is in the following section.

### 3 TRAFFIC GENERATION MODEL DESCRIPTION

We already pointed out that in order to drive conclusions not limited to the traffic characteristics of a particular LAN, we preferred not to directly perform simulations with measured traffic traces. Instead, we used a traffic model initially parameterised on the basis of measurements and then varied its parameters.

The model we implemented belongs to that category of bursty fluid models that try to achieve Self Similarity by aggregation of ON OFF sources with heavy tailed distributed on and/or off periods, as explained in (Morin, 1996) or in (Willinger, 1995). In particular, our model output consists of the rate generated by sources that can become active according to a Poisson process with parameter  $\lambda$ , when active generate traffic at a fixed rate  $R$  and whose active state duration  $h$  is heavy tailed distributed (see Figure 1). We will refer to that model as the Poissonian Arrival of Bursts (PAB) model. References to it can also be found in (Roberts, 1997).

Instead of investigating "a priori" whether the infinite source approximation were valid or not to correctly reproduce the traffic generated by a finite (and indeed rather limited) number of users/applications, we preferred to compare the queuing behaviour of real traces and simulated traces, usually finding a good match.



**Figure 1** – Traffic generation according to the PAB model.

The exact probability density function of the burst duration  $h$  is reported in equation (1), where  $H$  is the model's resulting Hurst parameter and  $T_c$  and  $\varepsilon$  have the meaning of maximum and minimum burst duration, respectively.

$$f_h(x) = \frac{1}{1 - \left(\frac{T_c}{\varepsilon}\right)^{2H-3}} \frac{1-2H}{\varepsilon^{2H-3}} x^{2H-4} \quad (\varepsilon \leq x \leq T_c, 0 \text{ otherwise}) \quad (1)$$

Due to the presence of a nonzero  $\varepsilon$  and of a non infinite  $T_c$ , the exact expression of the Index of Dispersion for Counts (IDC) for the model generated traffic drifts from the ideal one, which would be as in reported in (2) and correspond to the IDC of asymptotically Self Similar traffic. For any nonzero  $\varepsilon$  and finite  $T_c$ , the drift becomes more and more evident as  $t$  approaches zero or tend to infinity.

$$\text{IDC}(t) = Kt^{2H-1} \quad (\text{where } K \text{ is a constant}) \quad (2)$$

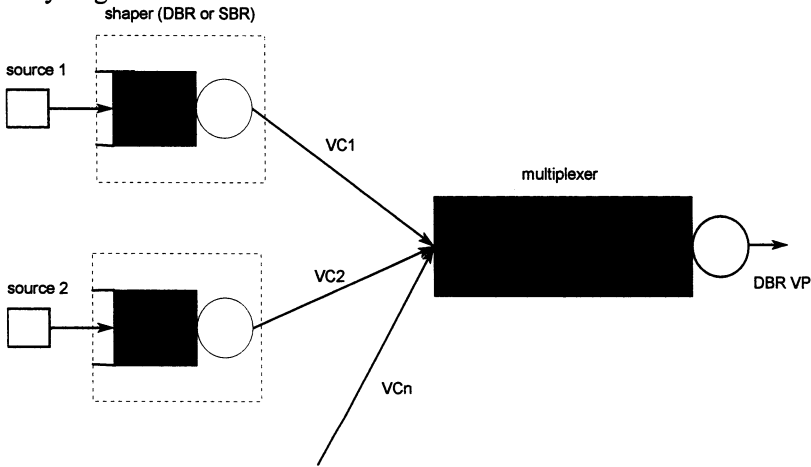
The model is thus characterised by five parameters:  $\lambda$ ,  $R$ ,  $H$ ,  $\varepsilon$  and  $T_c$ . We fixed the first three with the aid of three equations (not reported here) and of the three parameters extracted from measurements (mean rate, peak factor and Hurst parameter, see section 2), whereas  $T_c$  and  $\varepsilon$  can be considered as “freedom degrees” of the model. In finite time simulations, they must be set to values different from infinite and zero, respectively. We developed an algorithm that, given an interval  $[t_1, t_2]$  over which a “small drift” of the real IDC from the ideal expression reported in (2) is allowed, finds the best  $\varepsilon$  and  $T_c$  choices for a given maximum simulation time. In our study, we always required a “small drift” within the time range  $[0.1s, 50s]$ . The model then results as Self Similar and highly correlated (i.e. “Long Range Dependent” – LRD) at least within that range. In (Paxon, 1997), as an example, it is recalled that long term correlations of LAN traffic have frequently been observed “from hundredths of milliseconds to tens of minutes”. We are not so far from this range, as even if beyond 50s the IDC curve starts to drift from the ideal one, the traffic remains correlated well above this value.

#### 4 THE SIMULATION SCENARIO

In the following we assume that the reader is familiar with these concepts: ATM Virtual Path (VP) and Virtual Channel (VC) connections, ATM traffic contracts, Deterministic Bit Rate (DBR) transfer capability, Statistical Bit Rate (SBR) transfer capability, Peak Cell Rate (PCR), Sustainable Cell Rate (SCR), Maximum Burst Size (MBS). Definitions can be found in ITU-T recommendation I.371 (ITU-T I.371, 1996) or in ATM Forum traffic management specification 4.0 (ATMF TM 4.0, 1996).

To address some of the traffic engineering issues outlined in the introduction, we simulated a simple scenario, which is the multiplexing of several VC connections into a single DBR VP Connection. This may occur, for example, on an output port of an ATM switch that collects traffic from an edge ATM network or directly from ATM cards of upstream IP routers. The DBR VP connection is assigned a fixed amount of the bandwidth of the physical link, as well as a fixed amount of buffer space.

All the simulations presented in this study were performed at the fluid level, i.e. traffic sources produce as an output a sequence of couples like (Time Interval, Rate during Time Interval). The queues do not receive cells, but only the information about the intensity of an incoming workload and the duration of this intensity. Figure 2 summarises the simulation scenario we considered.



**Figure 2** - The simulation scenario.

Each VC connection has a traffic contract that can be either DBR or SBR, and a PAB model as described in section 3 generates the source traffic.

In this study we considered two cases:

- each VC connection can access all the buffer space and bandwidth reserved to the DBR VP (unshaped case), i.e. its traffic contract is either not controlled or traffic contract parameters are set to values that prevent shaping devices from performing significant actions on the incoming flow (e.g. if the traffic contract is DBR, the PCR of the connection is set equal to the line rate).
- each VC connection, before accessing shared VP resources, is shaped according to a traffic contract, that can be either DBR or SBR.

In the DBR case, a “fluid” shaping device can be thought as a queue served at the Peak Cell Rate (PCR) of the connection. In the SBR case, as a queue that can be served either at PCR or at a lower rate which is the Sustainable Cell Rate (SCR) of

the connection: the service rate is PCR as long as a token pool of size Maximum Burst Size (MBS) is nonempty, SCR otherwise. The token pool size is initially set at MBS, and it increases (or decrease) at a rate which is SCR minus the current rate of incoming traffic. The pool size can never exceed  $[0, \text{MBS}]$ . The size of the buffer in the shaping devices is set to infinite, i.e. no losses can ever occur in them. In both the unshaped and shaped cases the QoS class of the VC connections is QoS class 2, i.e. there is a commitment of a Cell Loss Ratio lower than a certain bound for each connection. In the following we suppose that due to the complete sharing of VP resources, achieving this commitment at the VP level (i.e. in the VP buffer) is equal to achieve it for the single connection. As VC shapers' buffers, when present, are of infinite size, the VP buffer is the only point along the connections where overflow can occur.

As pointed out in the introduction, although there's no explicit commitment for delays in QoS class 2, the network engineering should not enable delays to become intolerably high. In our simplified scenario, delays can occur both in the VP buffer and, if shaping is performed, in VC shapers' buffers. Therefore, the delay statistic we consider will be the sum of two terms: the mean delay encountered in the VP buffer and the average of the mean delays encountered in the VC shapers (if any).

Instead of taking the pure output of the PAB models, to speed up the simulations we slotted them into fixed intervals of duration 0.1s. The lowest time dynamics we will be able to observe is thus limited to this value. Time dynamics lower than 0.1s would anyway have been filtered due to the buffer size of the multiplexer, which was chosen considerably high (see later).

The sources we used are five different parameterisations of the PAB model presented in section 3. Table 1 summarises the main parameters for each one of them.

**Table 1** – Parameters of the Poissonian Arrival of Bursts (PAB) model used as traffic sources

	<i>Mean (byte/s)</i>	<i>Peak Fact (bytes/0.1s)</i>	<i>H</i>
Parameterisation 1	682000	21655	0.8
Parameterisation 2	682000	21655	0.9
Parameterisation 3	682000	21655	0.7
Parameterisation 4	682000	43310	0.8
Parameterisation 5	682000	10827	0.8

In particular, the first parameter set was derived from the analysis of measurements (see section 2). The others are one parameter variations from the first, to study what happens with increased autocorrelation (parameterisation 2), reduced autocorrelation (parameterisation 3), increased burstiness (parameterisation 4) and reduced burstiness (parameterisation 5). The variation in the autocorrelation was obtained by varying the Hurst parameter (the higher it is, the more autocorrelated the traffic), while the variation in the burstiness was obtained by varying the peak factor (the higher it is, the more bursty the traffic).

In all the performed simulations, all the multiplexed sources belonged to the same parameterisation, i.e. we didn't consider the case of mixed types of traffic sources. Note that the value of the Peak Factor at 0.1s did not come directly from the analysis of measurements (that for technical reasons were taken with a period of 1s), but was extrapolated from the value computed at 1s, as described in the following.

If the traffic has Self Similar characteristics over a certain timescale range  $[t_1, t_2]$ , then its index of dispersion for counts, over this range, has the expression reported in (2). The value of the Peak Factor at time  $t$  corresponds to  $IDC(t)$ . In our 1Hz frequency measurements we could verify Self similar Characteristics over a range  $[1s, t_2]$ , where  $t_2$  depended on the measurement's day but was always of the order of hundredths of seconds. We also estimated a Hurst parameter  $H$  close to 0.8, computed the peak factor at 1s (i.e.  $IDC(1s)$ ) and finally computed the value of  $k$  in equation (2). If the hypothesis that the traffic has Self Similar characteristics also on lower timescales is made, then the computation of  $IDC(0.1s)$  i.e. the Peak Factor at 0.1s is straightforward. This hypothesis is supported from a lot of empirical data analysed in several studies, see e.g. (Paxon, 1997). It wouldn't make sense to extend it to timescales lower than 0.1s.

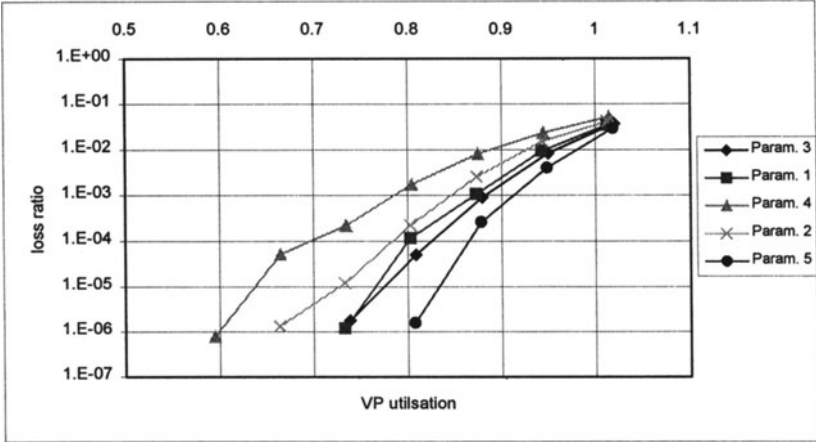
In each simulation run we considered at least a simulated time span of  $10^{+5}$  seconds (more than a day), in order to ensure the correct statistical behaviour of our sources, which have Long Range Dependent characteristics.

In all the simulations, the bandwidth of the DBR VP was fixed at 155 Mbit/s (thus representing the case of a single OC3 link dedicated to this kind of traffic) and its buffer space was fixed at 477000 bytes, i.e. 9000cells. This buffer space value is quite large (even if not unrealistic for today switches), and we choose it in order to better observe the effects of LRD traffic (the longer the buffer, the more relevant the impact of correlations properties in the traffic).

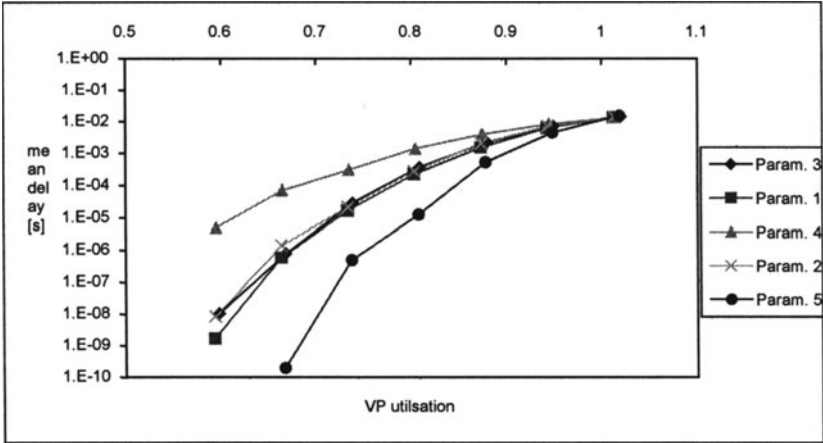
In Figures 3 and 4 we present the losses vs. utilisation and the mean delay vs. utilisation plots of the unshaped case for the five parameterisations of Table 1.

From Figure 3 it can be noted that the variation of the Hurst parameter (from 0.7 to 0.9) does not significantly impact the loss ratio in the VP buffer. This is partly due to the finiteness of the buffer. Performing other simulations with larger buffer space (may be unrealistic for an ATM switch) differences between the curves with different  $H$  value started to be more observable. However, it should not be deduced that Self Similar properties in the traffic do not affect performances, (indeed, in

Figure 3 there's no comparison with non Self Similar models). Only, whenever using heavy tailed ON OFF models in finite time simulations of finite buffer systems, the value of the Hurst parameter may not significantly affect the results. Mean delays are even less sensible to Hurts parameter variation (see Figure 4). On the contrary, the variation of the peak factor parameter, which is directly related to the burstiness of the sources on low timescales, significantly affects the performances, and should therefore receive much consideration when setting model parameters for engineering purposes.



**Figure 3** – Losses vs. utilisation plots for the five source parameterisations considered – all sources are unshaped.



**Figure 4** –Mean delay in multiplexer vs. utilisation plots for the five source parameterisations considered – all sources are unshaped.

It can be recalled that in order to meet a QoS commitment on CLR for multiplexed DBR or SBR ATM connections, four methods may be used.

The first one is simply to reduce the shared VP utilisation, i.e. to move the working point of curves like the ones of Figure 3 towards the left bottom corner. This is always possible and the multiplexing delay is even reduced, but it leads to lower incomes for the network operator.

The second one is to increase the level of multiplexing, i.e. to increase both the VP bandwidth and the number of admitted sources. Anyway, this is only possible if there are enough flows to multiplex, and in any case the VP bandwidth cannot exceed the one of the physical link. For some analytical considerations on the benefits of multiplexing with Fractional Brownian Motion traffic, see (Erramilli, 1996).

The third one is to increase the buffer space assigned to the VP. This always results in an increased multiplexing delay too, and there are cases where due to traffic source characteristics the buffer growth can be unacceptable. This is often referred as the “buffer ineffectiveness” for Long Range Dependent traffic.

The fourth one is to perform shaping on the flows before multiplexing according to some traffic contract parameters, that should be carefully chosen. Supposing to leave the VP utilisation the same, this always results in the creation of a second delay component (due to the queuing into the shaper’s buffers), while the delay component due the multiplexer is expected to be reduced. The effectiveness of such a method depending on traffic source characteristics has already been questioned in (Erramilli, 1996) and is further investigated in the following of this paper.

## 5 EFFECTIVENESS OF SHAPING IN REDUCING LOSS RATIO AND SIDE EFFECTS ON DELAYS

### 5.1 DBR shaping

We start considering the case of VC connections being shaped before multiplexing according to a DBR traffic contract: whenever the source has a peak whose intensity exceeds a given Peak Cell Rate, it is limited to that PCR and the excess work is buffered.

In the first simulated case, we multiplexed as many sources belonging to the first parameterisation listed in Table 1 as necessary to push, in the unshaped case, the loss ratio above  $10^{-5}$  (QoS class 2 target). This required 23 sources and led to a VP utilisation of 0.81.

While keeping the generated traffic the same, we then varied the PCR of the shapers in order to evaluate their effectiveness in reducing the loss ratio below the QoS class 2 target.

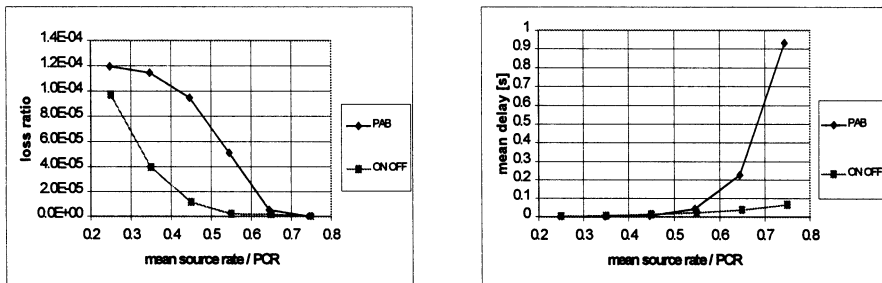
Results are reported in Figure 5 (for the moment, refer to the “PAB” plot only). In Figure 5(a) there is the value of the loss ratio in the VP buffer vs. the ratio of mean



source rate to PCR, while in Figure 5(b) the sum of the mean delay in the VP buffer with the average of the mean delays in the shapers' buffers vs. the same mentioned ratio. As the delay monotonically increases with a tighter shaping, it's clear that the increasing of the delay component introduced by the shaper always dominates the decreasing of the delay component in the VP buffer.

Also, it should be noted that losses start to differ from the unshaped case only when the shaping becomes "tighter enough" (e.g., referring to the PAB plot of Figure 5(a), only when mean source rate / PCR > 0.45) This means that the sharpest peaks, which are the only ones eliminated by a loose shaping, are not the main cause of losses in the multiplexer.

This result certainly depends on the rather big (9000 cells) size of the VP buffer, but also on the LRD characteristics of the PAB traffic sources. For comparison, always in Figure 5, we plotted the effect of shaping on "traditional" ON OFF sources with on and off periods exponentially distributed, with mean rate equal to the mean rate of PAB sources and peak rate during on periods equal to four times the mean source rate<sup>1</sup>. It's evident that in such a case even a looser shaping is more effective, and that a significant reduction in the loss ratio can be achieved without a dramatic increase in the delays.



**Figure 5(a) and 5(b)** – Effects on loss ratio and mean delay of DBR shaping for PAB parameterisation 1 sources (Table 1) and for ON OFF exponential sources. Abscissa values represent the ratio of the mean source rate to shapers' PCR.

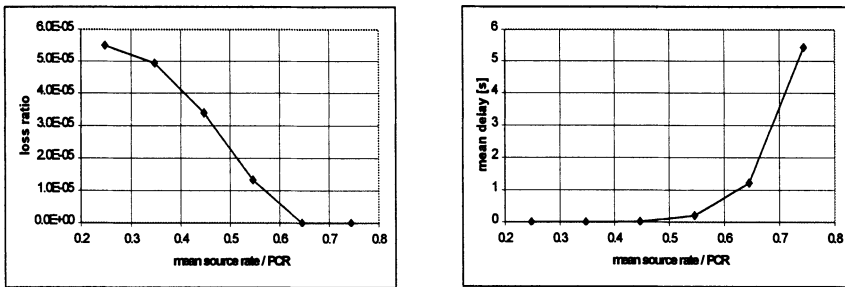
In Figures 6 through 9 we present plots analogous to Figure 5 for the other parameterisations of Table 1. The attempt was always to evaluate the effectiveness of the shaping to reduce the loss ratio from a starting value, in the unshaped case, above  $10^{-5}$ . In order to do so, for each parameterisation we varied the VP utilisation by adding or removing an integer number of sources, and this is the reason why the leftmost values in the loss plots may not be exactly the same.

Results show that the main drawback of performing DBR shaping on LRD traffic is the same outlined for parameterisation 1: the region where the shaping begins to

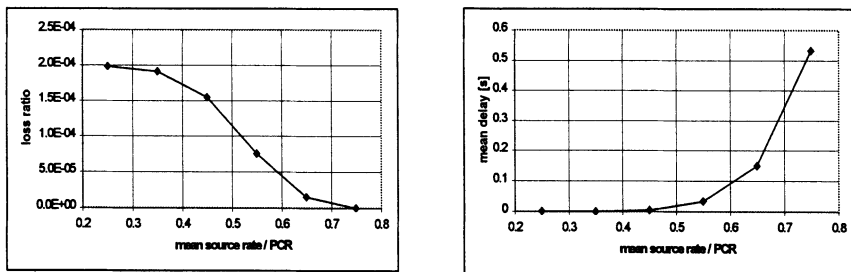
<sup>1</sup> Mean on and off period duration was chosen in order to obtain a loss ratio close to the one of the PAB sources in the unshaped case.

be effective on losses corresponds to the “sensible” region where mean delays in the shapers’ buffers start to (rather sharply) increase.

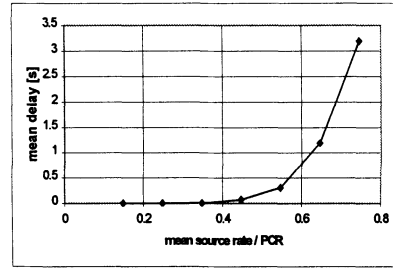
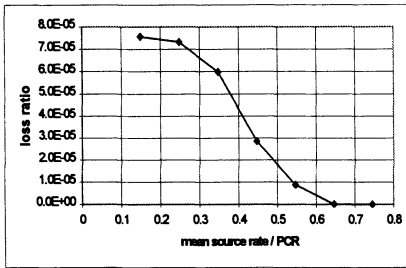
Moreover, it should be noted that for parameterisations 1, 2 and 3 (variation of the Hurst parameter, see Figures 5(b), 6(b) and 7(b)) the absolute values of delays for the same PCR are very different (the higher  $H$ , the higher the delays). We verified that these delays are dominated by the delay components introduced by the shapers. This means that when shaping at the source level, the value of the Hurst parameter plays a dominant role in the system performances. On the contrary, Figure 3 and even more Figure 4 showed that in an unshaped scenario the performance differences due to the Hurst parameter were negligible. So, another drawback of performing shaping is that more information about correlation properties of the sources (e.g. a reliable estimation of the Hurst parameter) would be needed.



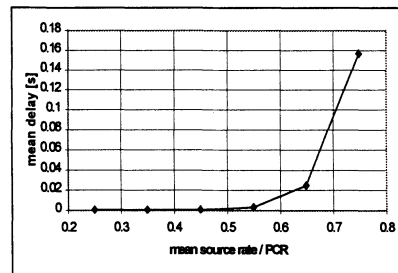
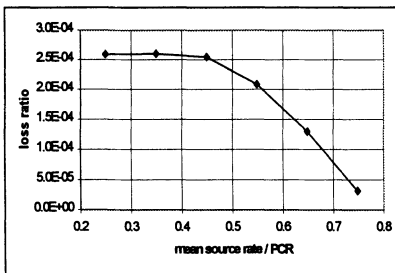
**Figure 6(a) and 6(b)** – Effects on loss ratio and mean delay of DBR shaping for PAB parameterisation 2 sources (see Table 1). Abscissa values represent the ratio of the mean source rate to shapers’ PCR.



**Figure 7(a) and 7(b)** – Effects on loss ratio and mean delay of DBR shaping for PAB parameterisation 3 sources (see Table 1). Abscissa values represent the ratio of the mean source rate to shapers’ PCR.



**Figure 8(a) and 8(b)** – Effects on loss ratio and mean delay of DBR shaping for PAB parameterisation 4 sources (see Table 1). Abscissa values represent the ratio of the mean source rate to shapers' PCR.



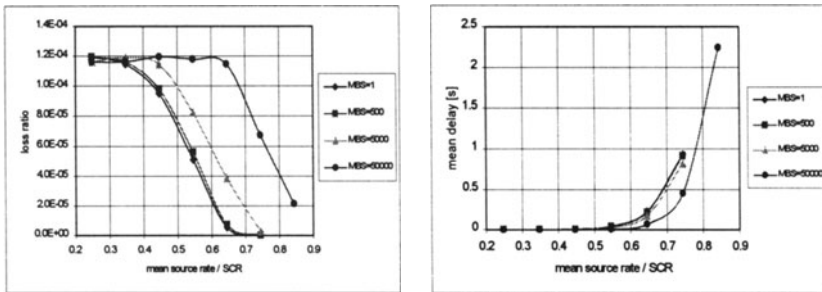
**Figure 9(a) and 9(b)** – Effects on loss ratio and mean delay of DBR shaping for PAB parameterisation 5 sources (see Table 1). Abscissa values represent the ratio of the mean source rate to shapers' PCR.

All these consideration lead to the conclusion that it's better not to alter the characteristics of the sources by performing a DBR shaping, (or at most perform only a loose, cautelative shaping), especially because the multiplexing gain can be considerable even with LDR sources (Erramilli, 1996).

## 5.2 SBR shaping

As the main drawback of DBR shaping seems to introduce significant delays in the shaper's buffer, we then investigated whether this could be overcome by using a more sophisticated shaping. The SBR traffic contract allows the source to transiently exceed a certain Sustained Cell Rate (SCR) rate, but on the long term it will be limited to it. The Maximum Burst Size (MBS) parameter limits the amount of this transiently generated excess traffic. Anyway, the source rate can never exceed a Peak Cell Rate (PCR). In order to limit the complexity of the analysis, we fixed the PCR at a rate higher than the maximum rate ever generated by a source. With such a choice, our implementation of an SBR shaper with a certain SCR and MBS = 1 cell is equivalent (for practical purposes) to a DBR shaper with PCR equal to this SCR.

We then let MBS assume the values 1, 500, 5000 and 50000 (in cells) and for each of them we performed the same analysis of the shaping effectiveness as before by varying the SCR value. In Figures 10(a) and 10(b) the results for parameterisation 1 are reported.

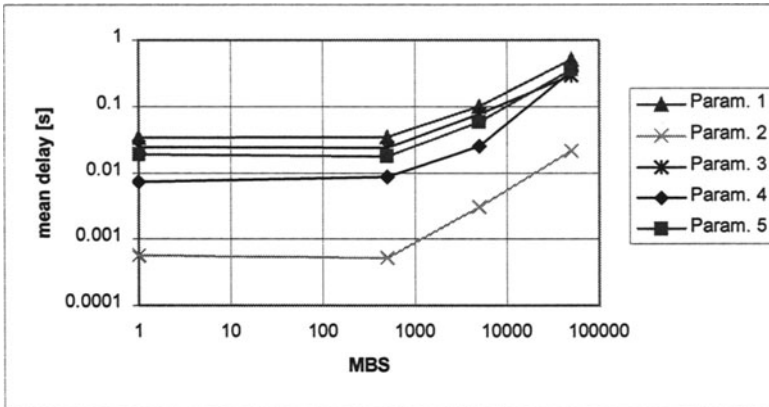


**Figure 10(a) and 10(b)** – Effects on loss ratio and mean delay of SBR shaping for PAB parameterisation 1 sources (see Table 1). Abscissa values represent the ratio of the mean source rate to shapers' SCR.

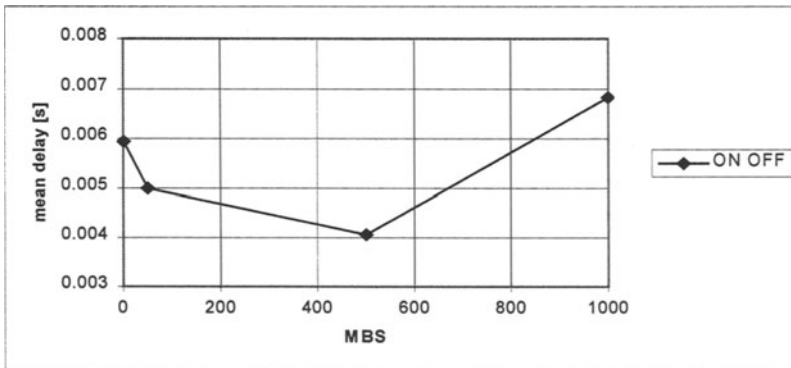
As expected, for the same SCR larger values of MBS lead to reduced delays (Figure 10(b)), as the main delay component (i.e. the one in the shapers) benefits for the increased “tolerance” of the shaping algorithm. Conversely, that tolerance throws into the VP buffer a traffic that is less filtered, and losses increase (Figure 10(a)). So, the right way to state whether SBR shaping is more or less effective than DBR shaping is to choose a loss ratio target value on Figure 10(a), read for each MBS curve what is the SCR needed to achieve it, and then find on Figure 10(b) the corresponding delay value. In Figure 11 we reported the results of such an operation for all the parameterisations considered.

As all the curves are monotonically growing, the conclusion is that SBR shaping is always less effective than DBR shaping (i.e. the MBS=1 point), probably because with these type of Self Similar sources no “typical burst length” can be identified. Note that absolute values of these curves should not be compared, as they depend also on the VP utilisation, which may differ from one parameterisation to another as explained before.

On the contrary, for the exponentially distributed ON OFF model considered before (see Figure 5), SBR shaping shows to be more effective than DBR shaping when MSB has a value around 500 cells (see Figure 12). The model had indeed an ON period average duration of 11ms, and an ON rate of 51471 cells/s: 500 cells is thus near to the “typical burst duration” of this type of source.



**Figure 11** – Effect on mean delay of SBR shaping for PAB parameterisation 1 through 5; varying MBS, for a fixed loss ratio target. Absolute values across the different parameterisations should not be compared.



**Figure 12** – Effect on mean delay of SBR shaping for an ON OFF exponential sources, varying MBS, for a fixed loss ratio target. – Note that the range of MBS values of interest is different from the one of Figure 11

## 6 CONCLUSION AND FUTURE WORK

In this paper we studied, by means of simulation, a simplified scenario where several VC ATM connections carrying QoS class 2 LAN traffic are multiplexed on a bandwidth resourced DBR VP ATM connection. Traffic was generated on the VC connections with a model having Long Range Dependent behaviour, parameterised on the basis of real aggregate LAN traffic measurements. The goal was to study the trade off between reduced loss ratio in the VP buffer (or, equivalently, increased VP utilisation) and increased transfer delays when the VC

connections were shaped according to some traffic contract. Results showed that when shaping tighter and tighter according to a DBR traffic contract, the reduction of losses in the VP buffer becomes significant only in the region where the mean delays in the shapers' buffers have started to increase, i.e. where the risk of having huge peaks in the delays is probably high (see Figures 5 through 10).

In addition, results showed that when shaping according to a SBR traffic contract, it's impossible to find a couple of parameters (SCR, MBS) leading to delays lower than the ones obtained with a DBR shaping, with the same loss ratio target level. This shows the difficulty to identify a "typical burst length" for these type of Long Range Dependent sources (Figure 11).

The results remain similar even if varying the parameters of the Long Range Dependent traffic generations models. On the contrary, they are rather different if other types of traffic generation models are used, such as a traditional ON OFF model with exponentially distributed ON and OFF periods. In this case DBR shaping is more effective in reducing loss ratio and its side effects (delay increase) are less disturbing (Figure 5). Always in this case, we showed that with an appropriate choice of MBS, SBR shaping can be more effective than DBR shaping, and therefore it makes sense looking for a "typical burst duration" (Figure 12). Unfortunately, LAN traffic characteristics are certainly closer to the ones showed by LRD models.

The relation between losses and delay curves, shaping parameters, traffic model generation parameters and buffer space was not addressed in this paper, but the results suggest some caution with setting tight shaping parameters until more light on the subject is shed.

## 7 ACKNOWLEDGEMENTS

The authors wish to acknowledge Mr. Damiano Inguaggiato and Luca Viale for cooperating in the measurement collection phase, and Mr. Marco Canova for writing part of the measurement analysis code.

## 8 REFERENCES

- ATM Forum TM4.0 (1996): Traffic Management specification - Version 4.0.
- Bonaventure, O. (1997) – A simulation study of TCP with the proposed GFR service category – Presented at Dagstuhl Seminar 9725 on High Performance Networks for Multimedia Applications, June 15-20, 1997, Schloss Dagstuhl, Germany.
- Erramilli, A. - Narayan, O. and Willinger, W. (1996) – Experimental Queueing analysis with Long Range Dependent packet traffic – IEEE/ACM Transactions on networking, Vol. 4 No.2 – pp. 209-223 – April 1996.
- Gusella, R. (1990) – A measurement study of diskless workstation traffic on an Ethernet. - IEEE transactions on communications – Sep. 1990
- ITU-T Recommendation I.356 (1996) – B-ISDN ATM Layer Cell Transfer Parameters.

- ITU-T Recommendation I.371 (1996) - Traffic control and congestion control in B-ISDN.
- Leland, W.E. and Wilson, D.V. (1991) – High time resolution measurements and analysis of LAN traffic: Implications for LAN interconnections – IEEE Infocomm'91, paper 11D.3.1
- Leland, W.E. - Taqqu M.S. - Willinger, W. and Wilson D.V. (1994) - On the self similar nature of Ethernet traffic - IEEE/ACM Transaction on networking, Vol 2, pp. 1-15, 1994.
- Morin, P.R. (1996) - The Impact of Self-Similarity on Network Performance Analysis – Publicly available on the WEB at <http://www.scs.carleton.ca/~morin/performance/selfsim/paper/paper.html>
- Paxon V. and Floyd, S. (1997) – Why we don't know how to simulate the Internet – Proceedings of Winter Simulation Conference – Atlanta 1997.
- Roberts, J. – Mocci, U. and Vitramo J. (1997) editors - Broadband Network Traffic - Final report of action COST 242 - Springer Verlag – Chapter 13.
- Willinger, W. - Taqqu, M.S. - Sherman, R and Wilson, D.V. (1995) - Self Similarity through high variability: statistical analysis of Ethernet LAN traffic at the source level. Proc. of SIGCOMM '95, pp.100 – 113.

## 9 BIOGRAPHY

Paolo Castelli received the degree in electronic engineering from Polytechnic of Turin, Italy, in 1985. After some years spent in the field of industrial automation, in 1989 he joined the Network Planning and Management department in CSELT, where he is now the head of the Traffic Engineering research unit. His area of activity includes the analysis of traffic management aspects over different networking technologies, with special attention to ATM networks. He participated to several EURESCOM and ACTS projects, and to the standardisation of the B-ISDN both in ITU-T SG 13 and in ETSI NA5.

Luisa Guida received the degree in telecommunication engineering from the University of Pisa, Italy, in 1997. During her thesis activity in CSELT, for which she received an award from the “Fondazione M. Bellisario”, she investigated on performance of ATM shapers in presence of Self Similar traffic. Currently she is a researcher of the Traffic Engineering research unit in CSELT and her interests include LAN traffic measurements, characterisation and modelling.

Maurizio Molina received the degree in electronic engineering from Polytechnic of Turin, Italy, in 1993. The thesis activity, for which he received an award from Telecom Italia, concerned a simulation study of connectionless services supporting through ATM. He has been working as a researcher in the Traffic Engineering research unit in CSELT since 1995, and his current activities are in the area of traffic modelling for management purposes. He participates to the standardisation of the B-ISDN in ITU-T SG 13 and in an ACTS project.

# Packet-based Approaches to ATM Cell Policing, and their Effects on Internet Traffic

*C. Song*

*Advisory Engineer, MCI*

*2100 Reston Parkway, Reston, Virginia, USA*

*+1 703.715.7114, +1 703.715.7066 (fax), csong@mci.net*

*R. Wilder*

*Senior Manager, MCI*

*2100 Reston Parkway, Reston, Virginia, USA,*

*+1 703.715.7114, +1 703.715.7066 (fax), wilder@mci.net*

*T. Dwight*

*Advisory Engineer, MCI*

*901 International Parkway, Richardson, Texas, USA*

*+1 972.498.1484, +1 972.498.1300 (fax), Tim.Dwight@mci.com*

## Abstract

IP traffic is commonly carried over ATM networks, using a Variable Bit Rate service. In this configuration, the ATM network typically measures the user's conformance to the subscribed traffic contract, using Usage Parameter Control (also known as policing) at the ATM network ingress. Traditional policing algorithms, which measure the user's rate of transmission at the ATM Layer, can lead to excessive loss when applied to packet data. This paper analyses several alternative policing algorithms, which attempt to minimise packet loss while still enforcing the ATM traffic contract. We observe that while some of the proposed algorithms perform better than others, all of them perform better (in terms of IP layer throughput) than traditional cell-based policing. Our study was conducted using traffic samples from the MCI Internet backbone. For this reason, our results may be more credible than those resulting the use of simulation or mathematical models.



## INTRODUCTION

Conventional ATM policing implementations are based on the Generic Cell Rate Algorithm (GCRA) defined in ITU-T Recommendation I.371 (I.371, 1995). To verify conformance with the traffic parameters associated with a given connection, they measure the interarrival times at the ATM layer (i.e., the gaps between the cells). In order to accommodate some measure of variability in this measured rate, they implement what is commonly termed a “leaky bucket”. The leaky bucket “drains” at the subscribed rate (PCR or SCR), and “fills” each time an inter-cell gap is judged to be too small. Cells are deemed to be nonconforming, and subjected to some form of punishment (generally either discard or tagging) if the bucket overflows.

Conformance checking as performed by the packet based policing mechanisms discussed in this paper, is based on both the leaky bucket fill, and on where in the encompassing Protocol Data Unit (PDU) the cells lie. In packet based policing, conformance checking is affected by and affects other cells in the same PDU.

The motivation for packet based policing is the fact that dropping one cell implies a loss of the entire packet to which the cell belongs (since the destination host will not be able to reassemble it). To maximise goodput for the higher layer protocol traffic, a packet based policer attempts to minimise the number of packets being policed, i.e., tagged or dropped, while still maintaining network performance and QoS commitments expressed in terms of cells.

The Usage Parameter Control definitions in (I.371, 1995) and (UNI 3.1, 1994) explicitly allow such behaviour. They require only that the *number of cells* judged nonconforming be within a certain tolerance of the number that would have been so judged by a “perfect policer”.

The packet based policing algorithms discussed in this paper mark as nonconforming, all cells subsequent to a nonconforming cell in the same packet. In standard (cell-based) policing, some of the latter cells will typically be judged conforming. Thus packet based policing reduces the number of nonconforming packets (packets in which one or more cell was judged nonconforming) by making room in the leaky bucket for subsequent cells belonging to new packets.

In this paper we analyse the effects of alternative policing algorithms, using traffic traces collected from the MCI Internet backbone, one of the largest, fastest, and busiest U.S. core IP backbones in existence. In the MCI Internet backbone, as in many carrier backbones, customer traffic is aggregated via a series of successively higher speed devices, into a set of core IP routers. The core routers are in turn interconnected via ATM PVCs, through an inner ring of ATM switches. The traces used in this study represent traffic on these PVCs.

Our analysis consisted of feeding this trace data to programs emulating the various policing algorithms, to observe how the associated flow would have fared, had it been subject to the associated policing mechanism. Our main contribution in this study is a set of quantitative results based on actual traffic measured in a production environment typical of IP over ATM in large networks today.

## Applicability

This study is applicable to IP-over-ATM networks which utilize policing. We note two significant cases in which this configuration is found:

- When the IP and ATM components belong to different administrations, policing will typically be applied by the ATM operator based on subscription agreements between the two. For example, the ATM network may be a public network, and the IP network may belong to an Internet Service Provider (ISP). In the U.S., such an arrangement is quite common.
- When the IP network wishes to offer differentiated service, for example by mapping the IETF's RSVP signaling to ATM signaling and thereby mapping IP flows to ATM connections, policing will typically be used to ensure that the QoS requested at the IP layer, can be guaranteed by the ATM layer.

## Structure of this paper

Section 2 describes the evaluated policing algorithms. In Section 3, we describe our data collection environment and tools. In Section 4, we discuss the results of our study. Section 5 summarises our work.

## PACKET BASED POLICING ALGORITHMS

We evaluated five policing algorithms: one for cell policing and four for packet based policing. The cell policing algorithm is based on the GCRA (I.371, 1995), (UNI 3.1, 1994), (TM 4.0, 1997). We classify the four packet based policing algorithms into fixed bucket size and variable bucket size, depending on their mode of restricting the leaky bucket size. One set, termed *FBP-like*, for **Fixed-size Bucket Policing**, defines conformance per the standard GCRA, up to the point that a cell is determined to be nonconforming. From that point until the last cell of the current PDU is received, all cells on the associated connection are considered nonconforming. Such algorithms ensure that the leaky bucket size is not violated.

The other algorithm, which we call *VBP* for **Variable-size Bucket Policing**, relaxes the bucket size restriction normally enforced by the GCRA. It allows the leaky bucket to fill beyond its configured size, in order to ensure that the same policing outcome is applied to all cells of a given PDU. If the bucket is over-full when the first cell of a new PDU is received, that cell and all subsequent cells of the PDU to which it belongs are judged to be nonconforming. In this algorithm, either all cells of a given PDU are judged conforming, or they are all judged to be nonconforming. Note that the maximum additional bucket size allowed by this algorithm is given by  $MTU_{cells} - 1$ , where  $MTU_{cells}$  is the Maximum Transfer Unit of the underlying connection, expressed as a number of cells. For a bursty traffic source, the bucket fill can oscillate around the original bucket size, with an overfill no more than  $MTU_{cells} - 1$ .

We chose to evaluate these two types of algorithms for several reasons. First, between them they encompass a wide range of algorithm behaviour. Second, they are similar to the two widely used packet based cell discard mechanisms, Partial Packet Discard (PPD) and Early Packet Discard (EPD). *FBP* is like PPD in that a negative policing outcome is applied to all remaining cells of the associated PDU. The *FBP* algorithms result in different policing outcomes being applied to portions of the same PDU; much as PPD results in the discard of partial PDUs.

*VBP* is like EPD in its application of the same policing outcome to all cells of a given PDU. Our definition of *VBP* differs from EPD in that it does not have a separate threshold besides the leaky bucket size, to trigger a change in policer state. Effectively, the *VBP* bucket size is the triggering threshold, and the extended *VBP* bucket size is the leaky bucket size plus  $MTU_{cells}-1$ .

*VBP* applies the same policing outcome to all cells of a given PDU, including the last cell. For the *FBP* algorithms, it is desirable to ensure delivery of the last cell in a PDU regardless of the policing outcome applied to the rest of the PDU, in order to retain the packet boundary indication. If the last cell in a PDU is not delivered, frequently 2 PDUs will fail to be reassembled successfully. This happens because the SAR process unwittingly attempts to reassemble cells of two PDUs into a single PDU. The resulting PDU will almost certainly fail either the length check or the CRC validation performed by the SAR layer at the receiving end system.

We evaluated three variations of *FBP*, which differ in their handling of the last cell in a PDU:

- *FBP-a* applies the same policing outcome to the last cell, as was applied to the other cells of the PDU
- *FBP-n* always finds the last cell of a PDU to be conforming
- *FBP-g* applies to the last cell of the PDU, the result of the standard GCRA policer.

## CELL TRACES AND MEASUREMENT ENVIRONMENT

We obtained cell traces using an internally developed OC3 monitor (Apisdorf, Claffy, Thompson and Wilder, 1996). The monitor is a PC based platform with two Fore Systems ATM interface cards. Each card has a general purpose microprocessor that performs cell timestamping, payload stripping and DMA transfer over the PCI bus to the host memory. One card was used for each direction of an OC3 link.

We collected several trace samples, each consisting of cell arrival records corresponding to several million cells arriving over the same OC3 link. Each record contained the header from one received cell, and a timestamp representing the time at which the last bit of the cell was received. Timestamps were estimated using a 25 MHz hardware clock on the ATM interface card.

## Measurement Accuracy

It was not possible to schedule the on-card processor each time a cell arrived, due to the overhead associated with processing interrupts. Instead, cells were read into an input FIFO, which was polled within a tight loop such that the per cell processing time was smaller than the OC3 cell inter-arrival time. The processor left this loop when the FIFO was empty.

Loop re-entrance delay plus the per-cell processing delay caused small inaccuracies in the timestamps assigned to the cells. To ensure the accuracy of our study, we analysed the worst cast deviation of this timestamp from the actual arrival time. Our traces show a minimum of 62 clock ticks' difference between the timestamps assigned to consecutive cells. The OC3 cell interarrival time is 2.87  $\mu$ s, or 71.9 ticks of a 25 MHz clock. Cells whose recorded interarrival times were smaller than this theoretical minimum, represent cases where the processor found multiple cells in the FIFO (62 ticks represents the processing time of a cell within the tight loop).

The longest sequence of cells with 62 ms interarrival times, in any of the recorded traces, was seven (7). Therefore the maximum error between the actual arrival time and the recorded arrival time, was as indicated in equation (1) below.

$$7 \times (71.9 - 62) = 69.3 \text{ ticks} \quad (1)$$

Note that this is less than one OC3 cell time (71.9 ticks). We conclude that the error introduced via our method of assigning timestamps, does not significantly impact the validity of our results.

## Network Configuration

We collected traces at the Washington DC POP on MCI's Internet backbone. We inserted the OC3 monitor into the OC3 link between an ATM switch and an IP router at this POP. The switch is connected to other switches across the country via OC3 and OC12 links. The router, considered part of the backbone core, forms the boundary between the ATM network and the IP access to the backbone. Connected to this core router over various media are many access routers that provide hundreds of customer connection ports. Ranging from DS0 to DS3 in speed, these ports connect MCI's customers including NAPs, regional networks, and corporate leased lines.

ATM cells traced by the OC3 monitor originate and terminate at core routers' ATM ports. Each VC seen by the OC3 monitor is backbone edge to edge, aggregating IP traffic to/from a large number of customers behind the access routers. In order to achieve maximum statistical multiplexing, the ATM interfaces on the core routers do not implement traffic shaping (i.e., they make no attempt to control their rate of transmission); nor do the ATM switches implement policing.

This study analyses what would have been the result, had the VCs in question been subjected to each of the proposed policing implementations. The traffic

parameters (SCR and MBS) of the VCs were varied over a range of possible values, and for each set of values, the traces were subjected to a simulation of each policing mechanism. Policing outcomes were collected and analysed, as discussed in the next section.

## EVALUATION RESULTS

We used three criteria to evaluate the packet based policing algorithms:

- The number of packets containing one or more cells judged to be nonconforming (for simplicity, referred to as *nonconforming packets*). To ensure high packet-level goodput, it is desirable to minimise this number.
- The ratio of nonconforming cells to nonconforming packets. It is desirable to maximise this number.
- Number of multi-cell packets in which the last cell was judged nonconforming but one or more of the previous cells was judged conforming. It is desirable to minimise this number, as such “partially conforming packets” can lead to a blurring of packet boundaries, and thus to the loss of multiple packets due to reassembly failure.

All the trace data exhibit essentially the same behaviour for all the evaluated algorithms. In the interest of brevity, we focus our presentation of results on one data trace, collected on November 6 1996 at about 13:00, containing 10 618 902 cells and 1 484 268 packets.

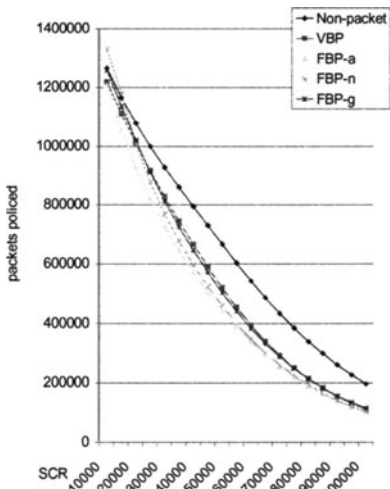


Figure 1 - packets policed (MBS = 50)

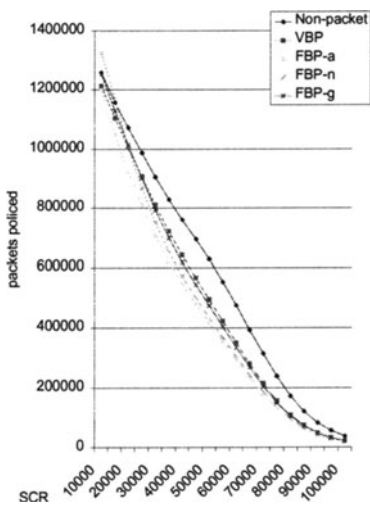


Figure 2 - packets policed (MBS = 200)

Figures 1 and 2 show the number of packets judged nonconforming, by each of the evaluated algorithms\*. Figure 1 is for Maximum Burst Size (MBS) set to 50 cells, and Figure 2 is for MBS set to 200 cells. The Y axis indicates the number of “policed packets”. The X axis indicates the Sustainable Cell Rate (SCR) setting, ranging from 10 000 to 100 000 cells/second with an increment of 5 000. The Peak Cell Rate (PCR) is in all cases assumed to be set to the OC3 line rate.

As expected, the cell-based UPC algorithm produces the highest number of policed packets. The *FBP-a* produces the least number of policed packets. All *FBP* algorithms produce lower numbers of policed packets than *VBP* does, except when the SCR is very low. Figures 3 and 4 suggest a possible explanation, showing the number of policed packets vs. the packet size (expressed as the number of cells in a packet). Both figures have MBS 200. Figure 3 is for the SCR setting of 10 000 and Figure 4 is the same for SCR=25 000. The graphs indicate that those algorithms showing fewer policed packets tend to show more policed packets of size 5 cells or greater, and far fewer policed packets of size less than 5 cells. When combined, the total numbers of policed packets are lower than those produced by algorithms that tend to police fewer large packets and more small packets.

---

\* The figures use the phrase “policed packet” to refer to a packet in which one or more cells was judged to be non-conforming.

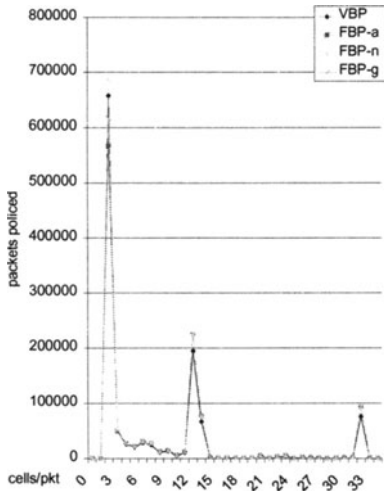


Figure 3 - policed packets vs. packet size (MBS=200, SCR=10 000)

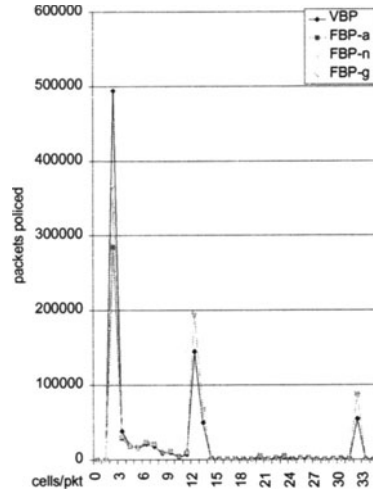


Figure 4 - policed packets vs. packet size (MBS=200, SCR=25 000)

We observe that the effects of packet based policing algorithms appear related to the IP packet burst distribution and size characteristics. Large packets will incur large emission delay, causing other packets to queue up behind them (so-called “ACK compression”, is an example of this phenomenon). Since most IP packets are small, such queued packets tend to be small. The result is a “packet train” consisting of a large packet followed by several smaller packets, which arrive back-to-back at the next multiplexing point in the network.

When such packet trains arrive, they cause different policing algorithms to behave differently. *VBP* is likely to find all cells of the large packet conforming, even if to do so it must allow its leaky bucket to surpass its configured size. Once the end of the large packet is detected, it may be necessary to mark several subsequent packets nonconforming in order to allow the leaky bucket to drain.

With the FBP algorithms, the rigidly enforced bucket size prevents the bucket from becoming over-full. As a result, fewer subsequent packets are found to be nonconforming.

Figures 3 and 4 show that when SCR is decreased, the number of small packets policed by the *FBP* algorithms increases. This effect is most pronounced with *FBP-n*. At the SCR setting of 10 000, the *FBP-n* algorithm policed more packets than *VBP*, perhaps due to *FBP-n* never judging the last cell of a PDU to be nonconforming.

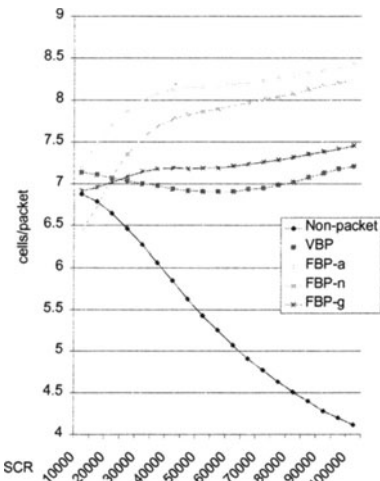


Figure 5 – policed cells per packet (MBS=50)

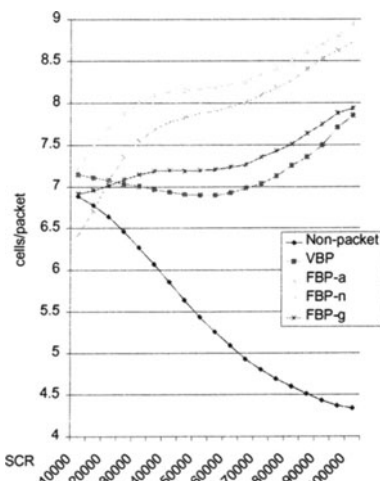


Figure 6 – policed cells per packet (MBS=200)

For our first evaluation criterion (minimisation of the number of policed packets), *FBP-a* performed the best, and the *VBP* and cell-based policing performed the worst. For the second criterion (maximisation of the ratio of policed cells to policed packets), *FBP-a* again performs the best, and *VBP* and cell-based policing perform the worst.

The third criterion was the algorithm’s ability to preserve packet boundaries. This criterion is applicable only to conventional cell-based policing, *FBP-a* and *FBP-g*. The other two, *VBP* and *FBP-n*, do not produce partial packets. Figures 7 and 8 show the numbers of partial packets produced with the MBS at 50 and 200 respectively. When the SCR setting is reasonably high, the *FBP-a* performs the best by producing the lowest number of partial packets. The cell-based policing algorithm produces the highest number of partial packets (i.e., performs the worst) while *FBP-g*’s number lies in between.



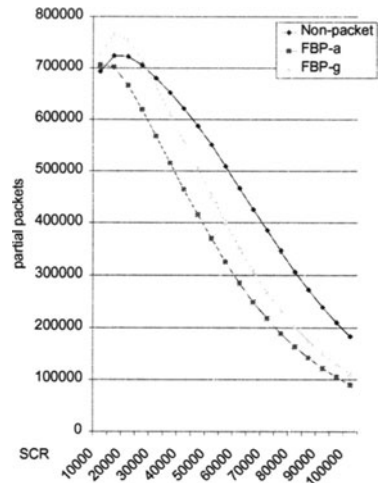


Figure 7 - partial packets (MBS=50)

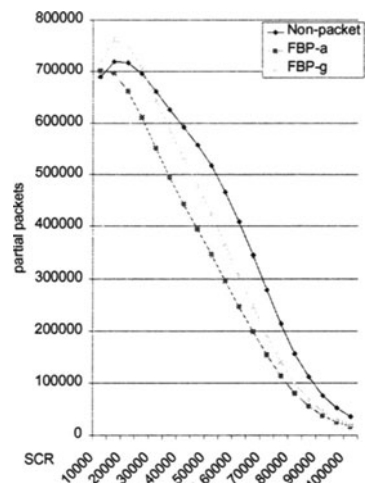


Figure 8 - partial packets (MBS=200)

The cell-based algorithm produces the lowest number when the SCR is set very low. We attempt to explain this by pointing out the effects of two factors: packet size and packet bursts. As we saw earlier, setting the SCR low leads to all packet based policing algorithms having the same number of policed packets of small size, but they still police a higher number of large packets than does the cell-based algorithm. With a low SCR, the effect of packet bursts intensifies such that the last cells of packets that would otherwise be judged as conforming (with a higher SCR), are judged as nonconforming.

The following table summarises the results.

Table 1 - Relative performance of policing algorithms (1 = best, 5 = worst)

	<i>Policing Algorithm</i>				
	<i>Cell-based</i>	<i>FBP-a</i>	<i>FBP-n</i>	<i>FBP-g</i>	<i>VBP</i>
Packets policed	5	1	2	3	4
Ratio of policed cells to policed packets	5	1	2	3	4
Preservation of packet boundaries	4	2	1	3	1

## SUMMARY AND FURTHER WORK

Our analysis of the 5 different policing algorithms using actual measured traffic suggests that packet based policing algorithms provide better service to IP data than does the traditional cell-based policer. Among packet based policing algorithms, the choice is between *FBP-a* and *FBP-n*. For reasonable settings of MBS and with the SCR set to above 25 000 for an OC3 line, *FBP-n* is a good choice, since it entails the second lowest number of policed packets, only slightly more than with *FBP-a*'s, but the algorithm preserves packet boundaries while *FBP-a* can not. With a low SCR, *FBP-n*'s number of policed packets becomes substantially higher and the trade-off between the number of policed packets and preservation of packet boundary is no longer straightforward.

In this work, we ignored the effects of packet sizes on policing effects. We hope to next develop a methodology to investigate this effect, so as to judge packet based policing algorithms as more of weighted sum (e.g., associate a weight with packet type and size) rather than a simple count. We will also examine the synergy between the packet-based policing and the packet based cell discard such as EPD/PPD with and without CLP=1 discard. We will need to gain a better understanding of the effects on policing by burst patterns by examining burst characteristics in greater detail.

## ACKNOWLEDGMENT

This material is based on work sponsored by the National Science Foundation, grant NCR-9321047. The very high speed Backbone Network Service (vBNS) project is managed and co-ordinated by MCI communications Corporation under sponsorship of the National Science Foundation. The Government has certain rights to this material. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## REFERENCES

- ITU-T Study Group 13 (1995), Recommendation I.371 "Traffic Control and Congestion Control in B-ISDN"
- The ATM Forum (1994), "ATM User Network Interface Specification, Ver. 3.1"
- The ATM Forum (1997), "Traffic Management Specification, Version 4.0"
- J. Apisdorf, K. Claffy, K. Thompson and R. Wilder (1996) "OC3MON: Flexible, Affordable, High Performance Statistics Collection", LISA X, September 1996

## BIOGRAPHIES

### Chuck Song

Chuck Song received his Ph.D. degree in Computer Science from the University of Wisconsin 1989. From 1989 to 1994 he worked in IBM. Chuck was on the IP router development team for the NSFNet project. Since 1995 he has been on the vBNS project engineering team in MCI Internet Engineering. Chuck has worked on the design of Concert Internet Plus as well as the engineering of vBNS.

### Rick Wilder

Rick Wilder is the senior manager of MCI's Internet Technologies group. He was one the original members of the design and implementation team for MCI's commercial internet backbone, and manages the engineering team for the very-high-speed Backbone Network Service (vBNS) project sponsored by the National Science Foundation. Rick has an MS degree in Computer Science from the American University.

### Tim Dwight

Tim Dwight is responsible for ATM network architecture and traffic engineering, for MCI's commercial ATM service. He holds an MS in Computer Science from the University of Kansas (1985) and holds several patents related to telecommunications.

# Optimising bandwidth reservation in IP/ATM internetworks using the guaranteed delay service

*C. A. Malcher Bastos*

*Dept<sup>o</sup> de Eng<sup>a</sup> de Telecomunicações,*

*Universidade Federal Fluminense*

*Rua Passo da Pátria 156, 24210-240 Niterói RJ, Brazil*

*telefax: +55 21 620 3935; e-mail: cmbastos@telecom.uff.br*

*M. A. Stanton*

*Dept<sup>o</sup> de Informática, Catholic University of Rio de Janeiro*

*Rua M. de S. Vicente 225, 22453-900 Rio de Janeiro RJ, Brazil*

*fax: +55 21 511 5645; e-mail: michael@inf.puc-rio.br*

*<http://www.inf.puc-rio.br/~michael>*

## **Abstract**

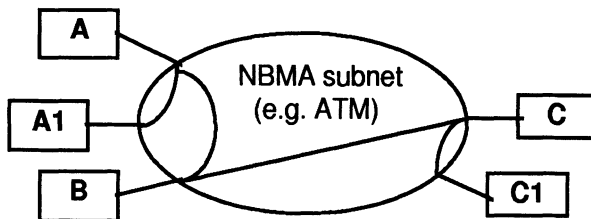
The service model proposed by the IETF for integrated services in IP internets, including the Internet, specifies, amongst others, a *guaranteed delay* service, which offers a deterministic delay for packet delivery. An analysis is presented of implementation aspects of this service, when ATM subnets are used. An evaluation is made of the reserved bandwidth necessary to guarantee a given maximum delay for a given application, taking into consideration alternative strategies for implementing IP over ATM, and factors such as traffic profile, end-to-end hop count, packet transmission with or without fragmentation, and bandwidth reservation for individual flows or groups of flows. It is shown that the reserved bandwidth necessary to guarantee a given delay is reduced by interleaving cells from different packets, by making flow group reservations, and by reducing the number of intermediate nodes. In large-scale internetworks implemented over (similarly large-scale) ATM subnets, switched virtual channels (VCs) have to be used. In such a situation, in order to perform flow group

reservations and to fragment packets, it has to be possible to interleave cells from different packets in the same virtual channel. VC management is also discussed, and the need is shown for procedures to optimise bandwidth and simplify the inclusion of flows in a group. A multilink procedure is proposed, which allows gradual alterations to be made to the bandwidth allocated to a flowgroup, thus permitting the inclusion and exclusion of individual flows in groups. An outline proposal is also presented of a SIA (Server for IP/ATM Integration), an ATM network device for setting-up and release of switched VCs used for traffic aggregation for support of IP flows.

## 1 INTRODUCTION

In the traditional internetworking architecture based on IP (Comer, 1995), the use of routers is essential for communication between two hosts belonging to separate IP networks, and consequently possessing different IP address prefixes, whereas two hosts in the same IP network can communicate directly. Communication between two stations connected to the same IP network obviously use this network's link-level addressing and protocol. It is therefore necessary to determine link-level addresses. In general, an address resolution protocol (ARP) is required to map an internet address to a link-level address. This type of protocol usually requires the use of broadcast communication facilities of the link-level subnet.

Certain subnet technologies, such as ATM and X.25, which provide a native capacity for routing, can be used to build extensive networks, known as "large clouds". A higher-level application can consider such a large cloud as a single link-level subnet. Frequently, on account of the size of the network, or of the technology itself, broadcast communication is unfeasible, making it impossible to use traditional address resolution mechanisms (ARP). Such networks are called NBMA (Non-Broadcast Multiple Access) (Braden et al., 1994b).



**Figure 1** Interconnection of routers in the traditional architecture.

Frequently, such networks correspond to more than one IP network. In such cases, the traditional model for communication between hosts on two such IP networks requires the mediation of one or more routers, even if the communicating hosts are connected to the same link-level subnet. In Figure 1, a

traditional IP architecture is assumed, with three routers, A, B and C, connected by IP networks as illustrated. A1 and C1 are hosts which share IP networks with A and C, respectively. Thus A and A1 share a common IP address prefix, as do C and C1. We suppose that these two IP networks are different. We have also supposed that the internet has been configured with an intermediate router, B, between A and C. Thus, should A1 need to communicate with C1, traffic between them will follow the route  $A1 \rightarrow A \rightarrow B \rightarrow C \rightarrow C1$ , since routers must be used for communication between hosts (or routers) with different IP address prefixes. However, at the link level, A and C are directly connected, and may communicate without the need to pass through B, in what is usually known as a cut-through or shortcut. (In this case, it is also possible to follow a shortcut from A1 directly to C1.)

ATM has come to be used in both local-area and wide-area networking, and in both public and corporate contexts. Since ATM is a NBMA technology, its use in communication subnets has brought about the development of appropriate standards for interoperating with TCP/IP protocols, even without considering its potential for integrating services.

These standards are RFC 1577 (Laubach, 1994), which describes an address resolution protocol in ATM subnets, RFC 1483 (Heinanen, 1993), describing encapsulation of datagrams when using AAL5, RFC 1626 (Atkinson, 1994), which defines the size of the maximum transmission unit (MTU), and RFC 1755 (Perez, 1995), dealing with set-up signalling for switched connections using UNI 3.1 (ATM Forum, 1995).

The solution based on RFC 1577 is known as “Classical IP over ATM”, and had as its primary objective the rapid deployment of IP internets using ATM subnets. In this solution, packets between hosts in different IP networks must pass through intermediate routers, even if the source and destination hosts are connected to the same NBMA subnet. There can only be direct communication between hosts that share the same IP address prefix. Such hosts are said to belong to the same LIS (Logical IP Subnet). Address resolution requires an ATM address resolution server (ATMARP) server, where all stations on the same LIS are registered. This form of packet forwarding completely ignores the communication subnet topology, increasing the hop count between source and destination, and requires packet reassembly in each router, with the introduction of increased delays. For exactly this reason, the Next Hop Resolution Protocol (NHRP) was proposed.

NHRP (Luciani, 1997) allows a host, which needs to communicate over an NBMA subnet, to determine the internet and link-level addresses of the subnet host (or router) nearest to the final destination. If source and destination belong to the same NBMA subnet, then the resulting address will be of the destination itself. If they belong to different subnets, the resulting address will be of the exit router to the subnet nearest to the destination. This protocol is not limited only to IP internets, and other groups (such as MPOA of the ATM Forum) are considering its use in other contexts.

These standards currently are designed for best-effort traffic, and make no allowances for real-time traffic or integration of services.

The inefficiency of Classical IP over ATM, doubts about the stability of NHRP (which we do not discuss here), and the large signalling overhead introduced by the use of Q.2931 (ITU-T, 1994) for setting up ATM virtual circuits, has led to the introduction of a new paradigm for routing IP packets in ATM subnets, known as *IP switching* (Callon, 1997). The basic idea is to establish a direct, albeit temporary, relationship between the internetwork and link levels, allowing packet forwarding based only on virtual channel identifiers, and using IP routing. In this fashion, the ATM switch assumes IP routing functions, making packet reassembly and NHRP unnecessary.

The TCP/IP architecture is evolving from a model of point-to-point delivery, with best-effort service, to multipoint delivery, with guarantees of quality of service (QoS) (Braden et al., 1994a). In order to offer performance guarantees, the flow profile of application traffic should be characterised by a token bucket, whose parameters are declared to the network routers, so that the required network resources may be reserved. Among the consequences of QoS guarantees are admission control and the introduction of flow state information in routers, which represent a considerable departure from the traditional model of IP internetworking.

The single most important service parameter to be guaranteed by the network is packet delay. Services currently being standardised include *guaranteed delay* and *controlled load* (Shenker, 1997; Wroclawski, 1996). The former provides deterministic guarantees of packet delay, and to implement this the router must implement an appropriate packet scheduling algorithm. The WFQ (Weighted Fair Queuing) (Demers, 1989; Partridge, 1994; Parekh, 1992) has been much studied and discussed by the working groups involved in developing these standards, and several variants have been proposed to support guaranteed service. This is the service which makes the greatest demands on network resources. There are also voice and video applications currently using the Internet, and these applications perform adequately when network load is low. The controlled load service was designed to cater for these applications, which do not need deterministic guarantees. This service supposes that applications will accept some loss of fidelity, in order to readjust their playback points as a function of the traffic.

The service model maintains the use of unreliable IP datagrams, and introduces dynamic resource reservations in routers which need to be continuously refreshed. Should there occur a routing alteration during a flow transmission, a reservation will be required for the new route, and this may cause a temporary interruption to the service. This is a consequence of the kind of routing used, which seeks the shortest route (in some metric) for each datagram. Current discussions include the use of QoS routing, which could result in offering as a solution a viable route, not necessarily the shortest one.

Today it is almost certain that it will soon be possible to implement services with controlled performance using TCP/IP. However, further study is needed to answer the following questions:

- Can we be sure that the model being proposed will allow the setting up of QoS control for a network as large as today's Internet?
- Supposing an affirmative answer to the first question, will the proposed model make efficient use of network resources, or will it be possible to find ways of further improving resource utilisation?

The present article is a contribution to answering this second question, when an ATM subnet is used for implementing an IP internet.

## 1.1 Organisation of this article

In section 2, we discuss the basic issues related to integrated services and to the model proposed by the IETF, which are relevant for our needs. Section 3 analyses the network bandwidth that must be reserved for a given application, in order to achieve a given delay when using the guaranteed delay service. The results of this analysis are applied to IP over ATM, and an alternative is proposed for the management of virtual circuits. Finally, section 4 presents the conclusions.

## 2 ISSUES RELATED TO THE SERVICE MODEL

### 2.1 The WFQ scheduling algorithm, and Parekh's thesis

The Weighted Fair Queuing (WFQ) algorithm was proposed by Demers (1989), based on a proposal of Nagle (1987), and uses bit-by-bit round robin scheduling, in order to guarantee an application fair access to the transmission bandwidth of a communications link.

Clearly it is not feasible to transmit the packets bit by bit, but, given the number of active flows and the size of a packet, it is easy to calculate the moment when the last bit of the packet would have been transmitted, if the transmission had been done bit by bit. Then the packets should be scheduled for transmission in accordance with the result of this calculation, and this would approximate asymptotically bit by bit round robin. One should also recognise that there will be occasional departures from the bit by bit model, caused by the arrival of a packet after the beginning of the transmission of another, which ought to have been transmitted afterwards.

Finally, it should be recognised that not all flows need to be allocated the same fraction of the transmission bandwidth. Supposing that there are  $N$  transmission queues, it is sufficient to subdivide the transmission bandwidth into cycles of  $M$  bits, where  $M \gg N$ , and allocate to each flow a bandwidth which is equivalent to the number of bits required. This particular variant is called WFQ.



Parekh's contribution (Partridge, 1994; Parekh, 1992) was to prove that, given a flow  $i$  whose traffic is shaped on the network boundary using a token bucket, rate-limited by a leaky bucket, and that WFQ is used in all nodes, the overall queuing delay of a packet belonging to flow  $i$  has a deterministic upper bound given by

$$D_i = \beta_i / g_i + (h_i - 1) \times l_i / g_i + \sum_{m=1}^i l_* / r_m \quad (1)$$

where

$D_i$  = maximum delay experienced by a data packet belonging to flow  $i$ ,

$\beta_i$  = bucket size,

$g_i$  = flow transmission rate, which should be greater than the rate of token generation

$h_i$  = hop count between source and destination,

$l_i$  = maximum size of a packet of flow  $i$ ,

$l_*$  = maximum packet size allowed in the network, and

$r_m$  = transmission rate for hop  $m$ .

WFQ is not the only kind of scheduling algorithm which can offer guarantees (Bennett, 1996; Georgiadis, 1996), but it is one of those that allows us to guarantee one of the smallest deterministic delays (Partridge, 1994). Recently, several variants of WFQ have been proposed, which are designed to reduce the cost of implementation, which is generally high.

## 2.2 The guaranteed delay service

The basic idea behind the guaranteed service is that a data flow can be described using a token bucket, and, with this description, a network element can calculate several parameters which will specify how this flow should be handled. Computing and combining the parameters from the different network elements along the route between the source and destination makes it possible to determine the maximum delay that a packet belonging to this flow will suffer when transmitted by this route (Shenker, 1997).

In order to offer delay guarantees, it is necessary to reserve network resources. To specify and make the reservation, an admission control mechanism is required, but the service specification does not deal with this. Additionally, making end-to-end guarantees requires the participation of all intermediate elements.

This service does not deal with the minimum or average delay, just with the maximum delay, not including the path latency, which must be added to the calculated queuing delay. If the application does not exceed the declared traffic parameters, the service also guarantees that packets will not be dropped because of buffer overflow.

The network element should make certain that the service offered is approximated by the fluid model. The fluid model of a service at rate  $R$  is equivalent to that offered by a dedicated link at rate  $R$  between the transmitter and the receiver, that is, the transmission is considered to be a continuous stream of bits. In this model, the service rendered is completely independent of all other flows.

Such a continuous stream of bits can be modelled by a token bucket with parameters  $(r, b)$ , where  $r$  is the rate of generation of tokens, and  $b$  is the bucket size, which is served by a link with transmission rate  $R$ . The definition of guaranteed service is based on the fact that the “fluid delay” of a flow, considered as a continuous stream of bits, is bounded by  $b/R$ , provided that  $R \geq r$ . Each network element must therefore assure that the queuing delay of any packet is bounded by  $b/R + C/R + D$ , where  $C$  and  $D$  describe the maximum local variation from the fluid model, since on a shared link it is not possible to transmit packets one bit at a time, simulating exactly the fluid model.

### 3 ALTERNATIVES FOR IMPLEMENTING IP OVER ATM

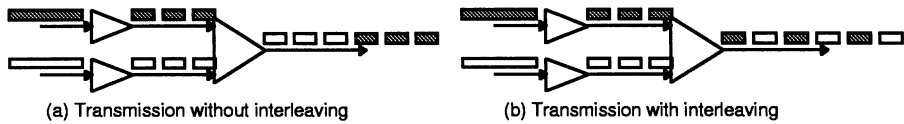
In order to implement the guaranteed delay service, it will be necessary to reserve transmission bandwidth along the entire route. To achieve this, the network elements should implement a packet scheduling algorithm which allows such guarantees to be offered.

In this section, a number of alternatives will be described for interconnecting routers using ATM virtual circuits, and these will be analysed in terms of the bandwidth which must be reserved in order to guarantee a particular maximum packet delay, supposing the use of the guaranteed delay service. These alternatives will be compared and related to proposals for IP over ATM.

The work reported in this section was motivated by discussions on the mailing list of the *intserv* working group of the IETF, following a contribution by Keshav (Intserv, 1996). Our principal contributions include the systematic analysis performed for the guaranteed service over ATM, the presentation of quantitative results, and the relationship between these results and some protocols proposed for ATM.

#### 3.1 Guaranteed delay service and packet fragmentation

By packet fragment, we mean a data unit capable of identification and individual forwarding, which may be transmitted interleaved with fragments of other packets. Analysing Figure 2, we note that non-interleaved transmission is equivalent to transmission without fragmentation, in the sense that an entire packet must be transmitted, before the transmission of another one may begin.



**Figure 2** Packet transmission with and without interleaving of fragments.

It is important to emphasise that in an ATM subnet a packet is always fragmented before transmission, since ATM uses fixed length cells. However, in this article, a packet is only considered to be fragmented when its cells may be interleaved with those of other packets in the same virtual circuit. In this sense, AAL5, for example, does not permit packet fragmentation at the VC level.

Equation 2, derived from equation 1, is used to calculate the bandwidth which should be reserved to guarantee a delay  $d$  as a function of the transmission parameters, when using WFQ.

$$g = (b + (N-1)P_1)/(d - (N.P_2/r)) \quad (2)$$

In equation 2,  $g$  is the bandwidth which must be reserved,  $b$  the token bucket size,  $N$  the end-to-end hop count,  $P_1$  the maximum packet size for the application using bandwidth  $g$ ,  $P_2$  the maximum packet size on the network belonging to other applications,  $d$  the desired maximum delay and  $r$  the transmission rate of the physical link.

The token generation rate does not appear explicitly in equation 2. For this equation to be valid,  $g$  must be at least as large as the application token generation rate. All hops are considered to support the same transmission rate.

The following subsections describe several different alternatives for the use of virtual circuits in implementing IP over ATM, relating them to equation 2. In a later section, this equation will be used to compare the bandwidth requirements of these alternatives for a given guaranteed delay.

### 3.1.1 *Guaranteed traffic aggregation and equivalent flows*

Independently of packet fragmentation, which will be analysed below, a further factor for reducing the required bandwidth for guaranteed service is flow grouping. Rampal (1996) has shown that, in certain cases, the required bandwidth for a flow group may be less than the sum of the individual bandwidths. In order to make a group reservation, it is necessary to choose a set of parameters which will be sufficient to cater for the whole group. Thus, considering the example of two flows we have

$$b_g = b_1 + b_2 \quad , \quad (3a)$$

$$\rho_g = \rho_1 + \rho_2 \quad , \quad (3b)$$

$$P_g = \max (P_1, P_2) . \quad (3c)$$

In the equations 3 above,  $b$  is the bucket size,  $\rho$  the token generation rate,  $P$  the maximum packet size, and  $g$  indicates the value to be used for the flow group (flow 1 and flow 2). It can be shown that in several cases the bandwidth to be reserved for the flow group with index  $g$  is less than the sum of the total bandwidth required for the flows individually (indices 1 and 2). For this result to hold, the bandwidth reserved for the flow group must not be less than  $\rho_g$ , the sum of the individual token generation rates.

### 3.1.2 Interconnection of routers using a single virtual channel

This is the traditional implementation of IP over ATM, proposed for best-effort traffic, with reassembly of packets in each router. If we suppose the use of this option for guaranteed traffic, the resource to be reserved is part of the bandwidth allocated to the VC. The VC must possess capacity sufficient for several guaranteed flows. Alternatively, one can maintain a small number of VCs between pairs of routers. This option is equivalent to interconnecting routers with a dedicated link. In this case, fragmentation of packets will not be caused by the ATM network, as the packets will be reassembled in each router, and will be transmitted serially, without interleaving (Figure 2(a)). Nevertheless, traffic can be aggregated, as proposed in (Rampal, 1996), which can contribute to reducing bandwidth requirements, as will be seen below.

To analyse this case, which corresponds to no fragmentation, we consider  $P_1 = P_2$  in equation 2, that is, maximum application packet size is equal to the maximum network packet size.

### 3.1.3 Interconnecting routers using dedicated VCs for each flow

In this case, an intermediate router receives the cells of a packet, analyses its header, makes a routing decision, and forwards the cells by a specific VC. Packets are encapsulated using AAL5, which implies in reassembly and refragmentation before forwarding. This case is identical to the previous one, except for the fact that cells from different packets can be interleaved, since each flow uses a separate VC. Nevertheless, all cells of a packet must be received by a router before retransmission can occur, and, in spite of the use of fragmentation, there is a reassembly delay at each network router. Since virtual circuits, resource reservation and traffic policing in ATM are performed end-to-end, there is no opportunity here for traffic aggregation or reserved bandwidth sharing, since the virtual circuits begin and end at each router.

To simulate this case, in equation 2,  $P_1$  was chosen to be equal to the maximum application packet size, and  $P_2$  equal to 48 bytes, the payload of an ATM cell. This takes into consideration the fact that a packet needs to be received entirely before being retransmitted, but that it is only necessary to await the transmission of one cell, before retransmission by the router, since cell interleaving is possible (using different VCs). The cell headers have not been considered, as they will affect equally all the cases we are considering.

### *3.1.4 Interconnecting routers using virtual paths*

In this case, bandwidth can be allocated for a virtual path (VP), and this can be shared between different flows, which are multiplexed using different VC identifiers. thus aggregating traffic. From the point of view of fragmentation, this case is equivalent to the previous one. The disadvantage of this alternative is the upper bound of 256 on the number of VPs in a UNI, which reduces its scalability.

### *3.1.5 Interconnecting routers using a cell-based interface*

Another alternative is an interface which functions without packet reassembly, instead processing packets cell by cell. The interface accepts cells from a number of incoming virtual circuits, analyses their content, checks their destination IP address, and takes a routing decision without packet reassembly. Cells are processed and forwarded to outgoing virtual circuits, whenever these are available. Reassembly delay is eliminated and all three alternatives previously mentioned can be considered: VPs, a single shared VC, and separate dedicated VCs.

- The use of VPs permit traffic aggregation, and fragmented transmission of packets. However the upper bound on the number of VPs available continues to hold.
- The use of a single shared VC also permits traffic aggregation. To benefit from fragmentation, packets should be transmitted with interleaving. Cells would be received, processed and forwarded directly to outgoing virtual circuits with interleaving, without awaiting the arrival of all the cells of each packet. Since cells are forwarded with interleaving, without packet reassembly and subsequent segmentation, sharing of a single VC requires a multiplexed identifier field (MID), as used in AAL3/4, which allows the receiver to separate cells from different packets.
- The use of a dedicated VC for each flow allows packet fragmentation without a MID, but, on the other hand, does not support traffic aggregation. An IP switch, based on ATM technology, and which does not use *VCmerge* operations (Callon, 1997), may be considered to operate with dedicated VCs.

From the standpoint of equation 2, all the cases discussed in this subsection are similar, and for the purpose of comparison the size of a cell payload is used,

since the ATM header is the same in all cases. To take fragmentation into account, we consider that all network packets have the size of an ATM cell payload, that is,  $P_1 = P_2 = P = 48$  bytes. In reality, the application packet, for which we reserve bandwidth, is composed of several fragments (in the traditional sense), which correspond to ATM cells. Thus the delay calculated using equation 2 using the above parameters is that affecting the transmission of just one packet fragment from source to destination. We have therefore to introduce a correction, to include the time required for all fragments to reach the destination, and make a small modification to Keshav's original equation (Intserv, 1996).

Suppose that a packet of size  $F$ , composed of fragments of size  $P$ , is transmitted at a reserved rate  $g$ . This packet will take  $F/g$  seconds to be transmitted. If the first fragment takes  $d'$  seconds to reach its destination, then for the maximum delay for this packet to be  $d$ , the delay for each fragment must be  $d = d' + (F-P)/g$ , which may be rewritten

$$d' = (d - (F-P)/g) . \quad (4)$$

$F$  is the maximum packet size produced by the application which needs to reserve bandwidth  $g$  in order to guarantee a maximum total delay  $d$ . The factor  $(F - P)$  supposes that  $d'$  already includes the arrival of the entire first fragment, that is, after  $d'$  seconds we must still await the arrival of the remaining fragments of the packet, except for the first fragment, which will already have arrived. If we now replace  $d$  in equation 2 by  $d'$ , and use equation 4, then, solving for  $g$ , we obtain

$$g = (b + (N-2)P + F)/(d - (N.P/r)) . \quad (5)$$

This guarantees that after  $d$  seconds, all fragments will have arrived, and the packet will be available for the destination application. Therefore, when considering the case of fragmented transmission, we will use equation 5 instead of 2.

### 3.2 A quantitative comparison of some alternatives

This section compares a number of alternative approaches for implementing the guaranteed delay service ATM, in terms of their required bandwidth requirements. Using equations 2, 3 and 5, the following pairs of alternatives are studied:

- individual flow reservation, with and without fragmentation;
- individual and group reservations without fragmentation;
- individual and group reservations with fragmentation;
- group reservations, with and without fragmentation.

Comparisons are made through the use of the gain,  $G$ , which is the ratio of the bandwidth reservations in the two alternatives considered for each case. In the case of fragmentation, we consider that we have an ATM subnet, that is, the fragment is always of the same constant size. Thus, parameters such as bucket size and packet size can be specified as an integer number of fragments. For purposes of comparison, this same approach is used for the case of no fragmentation. Thus we have:

fragment size = $P$	hop count = $N$
application packet size = $P_1 = mP$	transmission rate = $r$
network maximum packet size = $P_2 = wP$	required delay = $d$
bucket size = $b = kP_1 = kmP$	reserved bandwidth = $g$

Based on these definitions, equation 2 can be rewritten as:

$$g = \frac{kmP + (N-1)mP}{d - NwP/r} \quad (6)$$

and equation 5, which deals with the case of transmission with fragmentation, becomes:

$$g = \frac{kmP + (N-2)P + mP}{d - NP/r} \quad (7)$$

The denominators in equations 6 and 7 are similar, except for the factor  $w$ , the maximum cell count for network packets. Supposing a large-scale network using a 34 Mbps transmission rate and an end-to-end hop count of 10, the value of the fraction  $NP/r$  is 0.11 ms. If we consider an admissible delay of 100 ms, which is considered a reasonable value in discussions on IETF lists, we conclude that this factor is negligible in ATM subnets, where the largest cross packet size is a single cell ( $w = 1$ ). If, in the case without fragmentation, the size of a cross packet is equivalent to 100 cells, about 4.7 Kbytes, this fraction corresponds to 11 ms, still only 11% of the admissible delay. With higher transmission rates, the influence of this factor will be smaller. In any case, it will only not be negligible in cases of networks without fragmentation and with sufficiently large packets.

In order to simplify the analysis, and for the reasons just discussed, the denominators of equations 6 and 7 will be considered identical. Whenever this approximation is invalid, the effect will be to increase the advantage of fragmentation compared with transmission without fragmentation.

### 3.2.1 Individual flow reservation with and without fragmentation

The comparison is made using equations 6 and 7. Defining the gain  $G_{\alpha}$  as the ratio of the two left hand sides, after rearranging the terms and division of numerator and denominator by  $mP$  we obtain:

$$G_{cs} = \frac{k + N - 1}{k + (N-2)/m + 1}$$

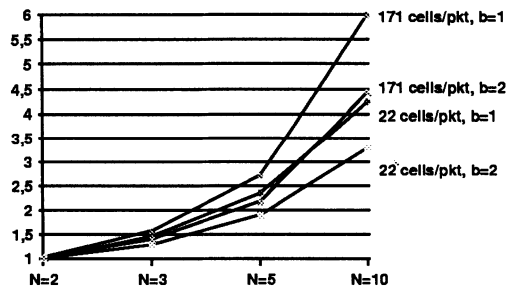
This expression allows some immediate conclusions. For single-cell packets,  $G_{cs} = 1$  (a trivial conclusion, since no fragmentation occurs). As the packet size increases ( $m \gg 1$ )  $G_{cs}$  tends to  $(k + N - 1)/(k + 1)$ , that is, the gain is directly proportional to the hop count,  $N$ , corrected by the bucket size,  $k$ . The larger the bucket size, the smaller the influence of the hop count. Thus, the effect of using a token bucket large enough to contain several packets ( $k \gg 1$ ) is to reduce the gain due to fragmentation.

For  $N \geq 2$ , we note that  $G_{cs} \geq 1$ . If packets are large ( $m \gg 1$ ), or if there are several intermediate nodes, the reserved bandwidth with fragmentation is always less than without. If an application uses a large bucket, the advantages of fragmentation are reduced, but we still have  $G_{cs} > 1$ , and fragmentation is always advantageous.

Figure 3 shows the dependence of the gain on the hop count, for packets of 22 and 171 cells (1 and 8 Kbytes), with buckets holding one or two packets. In the case of ATM, fragmentation can be achieved by one of the following alternatives:

- use of a VC for each flow,
- use of small packets, corresponding to a kind of pre-fragmentation, or
- use of an adaptation layer permitting cell interleaving in a single VC.

The influence of the hop count is clearly shown.



**Figure 3** Individual reservations with and without fragmentation.  
(Delay = 100 ms, Transmission rate = 34 Mbps)

### 3.2.2 Individual or group reservations without fragmentation

To analyse this case, we must compare the reservation required for  $z$  individual flows with that required for a single flow with parameters corresponding to the flow group. We shall analyse the case in which all the individual flows are described by the same parameters. In this case, the equivalent single flow uses a token bucket  $z$  times the size of the individual flow (see 3.3.1). The reservation for this equivalent flow is therefore obtained from equation 6, using a bucket size



of  $zb$ . To obtain the total reservation for  $z$  individual flows, we merely multiply equation 6 by the factor  $z$ . The resulting gain,  $G_{zis}$ , is given by

$$G_{zis} = \frac{zkmP + (N-1)zmP}{zkmP + (N-1)mP}$$

and, dividing numerator and denominator by  $zmP$ , we obtain

$$G_{zis} = \frac{k+N-1}{k+(N-1)/z}$$

From this expression, we conclude that group reservations, in the case of a set of similar flows, always reduces the bandwidth reservation, when compared with individual flow reservations ( $G_{zis} \geq 1$ ). Increase in bucket size tends to cancel the advantages of group reservations. The gain depends on the relation between  $N$  and  $z$ . For increasing values of  $z$ ,  $G_{zis}$  tends to  $(1 + (N - 1)/k)$ , and  $G_{zis}$  is proportional to the hop count,  $N$ , when  $z \gg N$ . When  $N/z \gg k$ , the gain tends to  $z$ . The influence of the hop count,  $N$ , increases in proportion to the increase in  $z$ , the number of flows in the group. The only restriction in this case is that the sum total of token generation for all flows must be less than the reserved bandwidth.

### 3.2.3 Individual or group reservation with fragmentation

In this case, similar reasoning is applied to 7, and the resulting gain,  $G_{zic}$  is given by:

$$G_{zic} = \frac{zkmP + z(N-2)P + zmP}{zkmP + (N-2)P + mP}$$

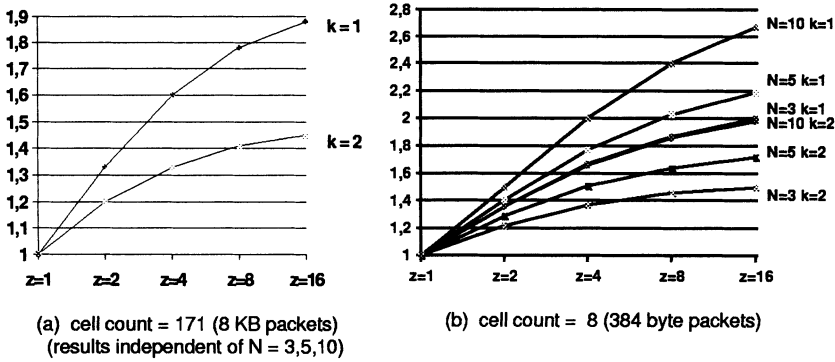
and, dividing numerator and denominator by  $zmP$ , we obtain

$$G_{zic} = \frac{k+1+(N-2)/m}{k+(N-2)/zm+1/z}$$

Note that  $G_{zic} > 1$ . In this case, we must consider the influence of the packet cell count,  $m$ . For small packets, the results are similar to the previous case. As packet size increases, the gain tends to  $(k + 1)/(k + 1/z)$ , a very different result from the previous case. The value of the gain is always between  $(k + 1)/(k + 1/2)$ , for two flows, and  $(k + 1)/k$ , for a large number of flows ( $z \gg 1$ ). For large packet sizes and  $k = 1$ , the gain varies between 1.33 and 2. The effect of increasing the bucket size is to reduce the gain. The gain also tends to  $z$ , with increase in the hop count. The main difference between this case and the case without fragmentation is the dependence of the gain on the packet size.

When the packet cell count is large (Figure 4(a)), the hop count has no importance. The larger the size of the flow group, the greater the advantages of fragmentation, but the influence of the bucket size is also increased. The figure shows buckets of one and two packets. For small packets (Figure 4 (b)), the hop count has a greater influence. The larger the hop count, the greater the

advantage of group reservation. With increase in bucket size, the advantages of fragmentation are reduced.



**Figure 4** Individual and group reservations with fragmentation.  
(Delay = 100 ms, Transmission rate = 34 Mbps)

### 3.2.4 Group reservations with and without fragmentation

In this case, the gain can be calculated from equations 6 and 7, using a bucket size equivalent to the number of flows in the group, that is,  $zkmP$ . We thus define the gain  $G_{zcs}$

$$G_{zcs} = \frac{zkmP + (N-1)mP}{zkmP + (N-2)P + mP}$$

and, dividing numerator and denominator by  $mP$ , we obtain

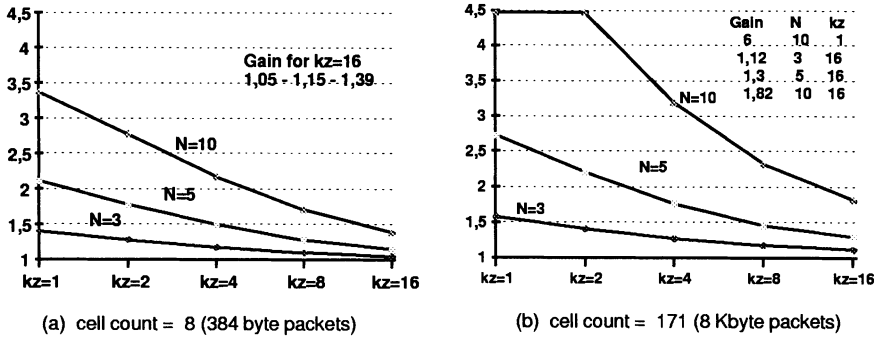
$$G_{zcs} = \frac{zk + (N-1)}{zk + (N-2)/m + 1}$$

The expression for  $G_{zcs}$  is identical to that for  $G_{cs}$ , the gain in the case of transmission with and without fragmentation, if we substitute  $zk$  for  $k$ . This is unsurprising, since group reservation is equivalent merely to increasing the bucket size. All the previous observations are directly applicable to this case. It is important to point out that, as the number of flows increases, the gain tends to 1. This means that for applications in which it is possible to group together a large number of flows, it is irrelevant, from the standpoint of bandwidth reservation, whether or not the network deals with packet fragmentation.

On the other hand, for applications with a limited number of transmitters, fragmentation can represent a considerable economy in bandwidth. For example, for  $m = 10$ ,  $k = 1$  and  $z = 4$  (or  $k = z = 2$ ), we have for  $N = 5, 7$  and  $10$ , respectively  $G_{zcs} = 1.5, 1.81$  and  $2.24$ , which represents in all cases a significant reduction in bandwidth reservation, even for cell counts of 10 (packets of 480 bytes). When  $m \gg N$ , the dependence on the hop count is linear, whilst if  $m \ll$

$N$  and  $N/m \gg zk$ , the gain tends to  $m$ . It should be remembered that increasing the cross packet size increases the advantages of fragmentation, by making the influence of network packet size non-negligible.

Figure 5 shows the dependence pointed out. For packets of 171 cells and a hop count of 10, we observe the great advantage of fragmentation, even for  $kz = 16$ , corresponding to 4 flows and a bucket size of 4 packets.



**Figure 5** Group reservations with and without fragmentation.  
(Delay = 100 ms, Transmission rate = 34 Mbps)

### 3.2.5 Conclusions from the analysis

#### Individual reservations:

- For a hop count of two, there is no advantage in fragmentation, independent of packet size - the gain is close to one. The reserved bandwidth depends strongly on the hop count in the absence of fragmentation. The reserved bandwidth depends directly on packet size for packets transmitted with or without fragmentation. This result can be obtained directly from equations 2 and 5;
- for a hop count of 10, the bandwidth reservation without fragmentation is at least five times as much as for a hop count of two, independent of packet size. Using fragmentation, the reserved bandwidth depends much less on the end-to-end hop count, but for smaller packets this dependence is still important;
- the larger the bucket size, the greater the bandwidth required for a given delay. The influence of bucket size does not depend on the hop count, and is greater when fragmentation is used. Large buckets reduce the benefits of fragmentation, but these benefits are still important;
- the larger the packet size, the greater the influence of fragmentation. For small packets, the gain is around 4 for a hop count of 10, whilst for large packets (cell counts of 171), the gain is around 6 for the same hop count (Figure 3). Thus we can conclude that packets should be fragmented, or the hop count be reduced. As in general we cannot reduce the hop count to 1 (possibly due to administrative restrictions which do not allow a shortcut from

source to destination), fragmentation should always be considered for the case of guaranteed traffic.

- The gain resulting from fragmentation is marginal, when packet reassembly is required in the router. This will only be advantageous when packets are larger than 4 Kbytes and the end-to-end hop count is at least 10. This case was not discussed here, but can be found in (Malcher Bastos, 1997).

### *Group reservations*

As resource reservation and traffic policing in ATM subnets are based on virtual circuits, one should evaluate the benefits of grouping several flows using the same virtual circuit. In this case, we have to consider an equivalent flow, whose parameters are obtained from equations 3. The only restriction is that the reserved bandwidth must exceed the sum of the separate token generation rates for the flows in the group.

- For flows with identical parameters, group reservation is always beneficial, from the standpoint of economising bandwidth, independently of packet size, hop count, flow group size or the use of fragmentation; it is important to compare reserved bandwidth for a group of flows, for the cases of transmission with and without fragmentation. This comparison indicates whether there is any advantage in transmitting packets with cell interleaving, when using a single VC. If the required bandwidth is similar in the two cases, then there will be little point in using fragmentation and, consequently, cell interleaving. In this case, flow aggregation provides the same benefits as fragmentation;
- group reservations reduce the advantages of fragmentation, since the gain due to fragmentation is reduced in proportion to the increase in token bucket size. If the group is large, it may be unnecessary to fragment, but group reservations are always beneficial;
- from the equation for gain, we see that the gain for a hop count of 2 is equal to 1. From Figure 5 we observe that the gain increases with increasing hop count, and decreases when more flows are included in a flow group. The gain also depends on the packet size. For instance, for a hop count of 10 and  $kz = 8$ , the gain is around 2, both for large packets (2.32) and small packets (1.71). But for  $kz = 4$ , the variation is between 3.2 and 2.2. As a rough approximation, we may say that, independently of packet size, for hop counts between 3 and 5, fragmentation is beneficial for flow groups with  $kz \leq 8$ . For hop counts between 5 and 10, fragmentation is still advantageous for  $kz > 16$ .

### **3.3 Consequences for IP over ATM and the Guaranteed Service**

From the standpoint of bandwidth reservation, there are advantages in the use of small packets, if we ignore the effect of increased header overhead. If an application generates large packets, these should be fragmented before

transmission. Small packets at the IP level can be produced by adequate definition of the MTU (maximum transmission unit). If the underlying subnet is cell-based, as in the case of ATM, then the packets can be fragmented at the subnet level, and this taken into account in calculating resource reservations.

An important result is that the interface of the router with the ATM subnet should be capable of processing and transmitting packets cell by cell. Such an interface initiates and terminates virtual circuits, processing packets cell by cell, and making possible the benefits of fragmentation. The technique of IP switching (Callon, 1997), which can associate a dedicated virtual circuit to a specific data flow, when based on ATM, automatically allows packet fragmentation. The IP switching techniques were not developed with this in mind, but it is an performance argument in their favour (Malcher Bastos, 1997). Should the VCmerge operation (Callon, 1997) be used, in order to support fragmentation one must use VPs (VPmerge), or a form of relating cells to a packet or to a transmitter, equivalent to the AAL3/4 MID.

It is important to reduce the end-to-end hop count, as the reserved bandwidth depends directly on this factor. This favours the use of NHRP, which is important for reducing transmission bandwidth, as well as reassembly latency at intermediate routers, which was one of its original objectives. It is interesting to note that we have so far not encountered a single IP/ATM document from the IETF or MPOA document from the ATM Forum, which refers to this matter or points out such benefits for ATM-based IP switching.

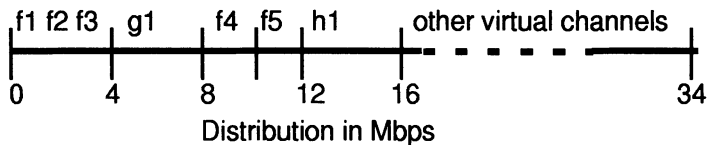
Transmitting packets belonging to different flows on the same VC or VP brings already mentioned benefits of bandwidth reduction through traffic aggregation. If VCs are used, fragmentation requires cell interleaving, and we need to know to which packet each cell belongs. One could use AAL3/4, or an equivalent scheme. This will increase the transmission overhead, but even so fragmentation is beneficial.

### 3.4 Management of virtual circuits

Several documents relating to IP over ATM (Berson, 1997; Borden, 1997; Berger, 1996), produced by the ISSLL (integrated services over specific link layers) working group of the IETF, refer to the management of virtual circuits and traffic aggregation as related items having high priority for further study. One aspect of virtual circuit management is traffic aggregation in a single virtual circuit. The use of a single, high capacity virtual circuit for different flows has the advantage of reducing the problem to one previously solved: the interconnection of routers by high capacity links (Berson, 1997). The problem of bandwidth management would be dealt with at the internet level, which would distribute bandwidth between different service types and specific applications.

Figure 6 shows qualitatively some aspects of bandwidth management of a 34 Mbps link between two network elements, such as used in connecting two routers using several ATM virtual circuits. We consider one VC of capacity 16 Mbps,

which is subdivided as follows: 4 Mbps are being shared for flows of type *f*, through a group reservation for the flows *f1*, *f2* and *f3*. Individual reservations have been made for flows *g1* and *h1*. Flows *f4* and *f5*, in spite of being of type *f*, have received individual reservations of 2 Mbps. From the figure it should be noted that, if a group reservation had also been made for *f4* and *f5*, it would still be possible to include in the group a further flow of type *f*. In such a situation, cell interleaving would have to be used to maintain packet fragmentation in the VC. Another option would be to use a dedicated VC for each separate flow, but in this case group reservations would not be possible.



**Figure 6** Distribution of data flows in a virtual channel.

In order that a virtual circuit (VC or VP) be set up in an ATM subnet, an end-to-end signalling procedure is required to negotiate parameters and reserve resources in network switches. One advantage of traffic aggregation in a shared virtual circuit is to eliminate, or at least reduce, the amount of signalling overhead.

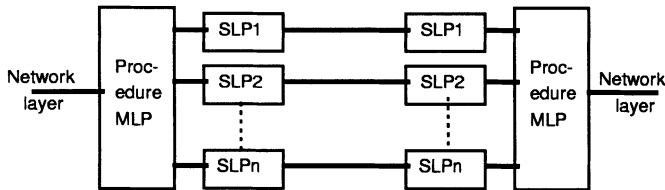
A disadvantage of use of a single high capacity VC, as described here, is that its management may be complex. If a large bandwidth were allocated to a given channel, this might be underused, producing exactly the opposite of the desired effect. On the other hand, if the reserved bandwidth of a shared VC is already totally used up, and a new flow were to be set up, this flow would not be able to use the existing channel, as the resources allocated would be insufficient. Two solutions are available: either we make an individual allocation for the new flow, with loss of the capacity to exploit traffic aggregation, or we allocate a larger bandwidth for the now larger flow group, and transfer the flows of the old group to this new channel, freeing the existing one.

One difficulty here is the possibility that resources are not available for reallocating a channel for the larger flow group. To avoid undesirable service interruptions, we should first allocate the new channel, and then transfer to it the existing flow group, before releasing the old channel. Obviously, this would not be possible, if the new channel's capacity exceeds the available bandwidth. The alternatives left open to us include: an individual reservation for the new flow; the use of a new route, refusing a flow which could be handled on this one; or the release of the existing shared channel, prior to establishing the new channel, with consequent service interruption, unacceptable here.

Thus we see that traffic aggregation on a single virtual circuit is by no means as simple as perhaps appears at first.

### 3.5 Multilink procedure for ATM: an alternative for VC management

A further alternative solution to the problem posed at the end of the previous section would be renegotiation of virtual circuit parameters during the phase of data exchange. If it were possible to alter these parameters, to cater for the new flow, the problem would be solved. However, this facility is not currently supported in ATM (ATM Forum, 1995a, 1995c, 1996). The relevant ITU-T recommendations (ITU-T, 1996<sup>a</sup>, 1996b) do not deal with renegotiation during the phase of data transmission.



**Figure 7** Multilink Procedure.

Another alternative to solve this problem would be a procedure analogous to the multilink procedure proposed for the link level of the X.75 protocol (ITU-T, 1993a). In this procedure, illustrated in Figure 7, there may exist several parallel physical transmission links, each one of which using as link-level protocol an instance of LAP-B (ITU-T, 1993b), called Single Link Procedure (SLP). Between the network layer and the SLP, a Multilink Procedure (MLP) is used to distribute among the different alternative links the packets to be transmitted. The MLP encapsulates these packets in a multilink frame, which includes a sequence number field, to allow the MLP layer to deliver the packets in the correct order.

In the case of ATM, a set of VCs connecting the same endpoints could be used in a similar way by a multilink procedure. Suppose, for example, that an 8 Mbps shared channel were implemented by a multilink group of four 2 Mbps channels. If it were necessary to increase the capacity of this multilink channel to 10 Mbps, it would suffice to add a further 2 Mbps channel. The difference between this procedure and the use of separately allocated channels is in the form of packet transmission. In the case of separately allocated channels, all cells of the same packet follow the same VC. In the multilink procedure, each time a packet is transmitted, its cells will be distributed amongst all channels of the multilink group, fully utilising the capacity of the group. As new flows are included or excluded, new virtual channels can be established or old ones released, permitting operations of traffic aggregation, which are not possible using dedicated channels.

The cells of a packet will need to be identified with a MID and a sequence number, and then distributed among the different VCs for transmission. On

arrival, cells will be regrouped by MID and reordered by sequence number. Among the duties of the multilink procedure will be the management of MIDs. Since cells of a single packet will be transmitted in different VCs, the MID will have to be unique within a VC group and not just within a single VC, as is traditional in AAL3/4, for example.

The multilink procedure is equivalent to transmission in a single channel, in the sense that the final bit of a group of packets sent by a group of  $k$  channels of bandwidth  $g/k$  will arrive at the same instant as would be the case if sent by a single channel of bandwidth  $g$ . This is easy to see, if we recognise that, although the transmission rate is reduced by a factor of  $k$ , the number of cells in each channel is similarly diminished.

In (Malcher Bastos, 1997), a description is given of a segmentation and reassembly sublayer, which allows the implementing of a multilink procedure. There is also a discussion of the feasibility of such an implementation, including the cost of performing the necessary additional functions, such as management of multiple VCs, distribution of cells on transmission, and their reordering on arrival. For the multilink procedure to be feasible, there must also exist a simple way to allocate bandwidth for a channel group, and that this effort be rewarded in terms of bandwidth economised. This point is also discussed, and a proposal is presented for the introduction of a connectionless service in ATM subnets, incorporating the mechanisms described in this section, which may be utilised for implementing the guaranteed delay IP service over ATM (Malcher Bastos, 1998).

This connectionless service may be implemented with the assistance of a device which we call a SIA (Server for IP/ATM Integration), which is also discussed in (Malcher Bastos, 1997). This device manages the setting up and release of switched virtual channels, permitting traffic aggregation in a single virtual channel, simplifying network resource management and permitting the reduction of bandwidth required for guaranteed service. A SIA only concerns itself with those matters related to the communication subnet, relegating to the IP layer those matters traditionally handled by connectionless servers.

A SIA is an ATM network device, and can simultaneously attend to several end systems and routers, belonging to different LISs. There may be several SIAs in the same ATM subnet, and each of these can operate independently of other SIAs connected to the same network, exercising only local functions. Alternatively, several SIAs may cooperate, communicating through shared SVCs, using cell interleaving and shortcuts, and mapping IP integrated services parameters onto ATM parameters.

A SIA can operate under the control of a router, setting up virtual circuits of specific IP flows, and mapping input channels into output channels, without the necessity for header analysis of each packet processed. This procedure is analogous to an IP switch, which *redirects* IP flows, binding them to specific virtual channels. In a SIA this redirect operation can also be performed locally, avoiding the unnecessary passage through a router of certain packets.



Thus, in order to carry out its functions, a SIA needs to resolve IP addresses using classical IP over ATM, use shortcuts, based on NHRP, which reduces the number of hops between source and destination, perform traffic aggregation on a single virtual channel, using cell interleaving, based on a SAR sub-layer developed specifically for this purpose, and perform redirection of IP flows, as well as traditional redirection, operating in this case like an IP switch.

#### 4 CONCLUSION

We have analysed some aspects of the interoperation of IP internets with ATM subnets, in relation to optimising the bandwidth allocated to an application using the guaranteed delay service in the integrated services model for IP internets. For such a service, the network elements should implement a packet scheduling algorithm based on some variant of WFQ. It is shown here that, in the case of the guaranteed service, if packets are fragmented before transmission, a considerable economy can be made in the amount of network resources which must be reserved, for a given maximum delay. The reserved bandwidth also depends on the end-to-end hop count, and although this dependence is stronger if fragmentation is not used, it still exists even with fragmentation. This result points out clearly the importance of NHRP, even if the network utilises IP switching techniques, which were originally proposed as an alternative to NHRP. Flow group reservations, instead of for individual flows, are another factor which can substantially reduce the reserved bandwidth.

Aspects of VC management were also discussed. Traffic policing in ATM can be performed for VCs or VPs. Using a VP to multiplex flow groups in a UNI introduces scaling problems, since one VP is required for each flow group. Alternatives include sharing a single high capacity VC, or renegotiating the traffic parameters of a connection. Both of these have drawbacks, and we introduced as another alternative the multilink procedure, proposed in (Malcher Bastos, 1997). This procedure allows a finer granularity to be used in bandwidth reservations, without losing the possibility of being used for flow groups. Finally, an outline proposal has been presented of a SIA (Server for IP/ATM Integration), an ATM network device for setting-up and release of switched VCs used for traffic aggregation, typically under control of an IP router, drawing together in one place various of the topics discussed in this article.

#### 5 REFERENCES

- Atkinson, R. (1994) *Default IP MTU for use over ATM AAL5*. RFC 1626.
- ATM Forum (1995a) *UNI Specification 3.1*. Prentice Hall, New Jersey.
- ATM Forum (1995b) *Traffic Management Specification V. 4.0*.
- ATM Forum (1995c) *Lan Emulation 1.0*.
- ATM Forum (1996) *User Network Interface (UNI) Signalling Specification V 4.0*

- Bennett, J. and Zhang, H. (1996) Hierarchical Packet Fair Queueing Algorithms. *Proc. ACM SIGCOMM 96*.
- Berson, S. and Berger, L. (1997) *IP Integrated Services with RSVP over ATM*. Internet Draft draft-ietf-issll-atm-support-03.txt.
- Berger, L. (1996) *RSVP Over ATM: Framework and UNI 3.0/3.1 Method*. Internet Draft draft-berger-rsvp-over-atm-00.ps.
- Berson, S. and Vincent, S. (1997) A "Classy" Approach to Aggregation for Integrated Services. Internet Draft, draft-berson-classy-approach-00.txt.
- Borden, M. et al. (1996) *Issues for RSVP and Integrated Services over ATM*. Internet-Draft draft-crawley-rsvp-over-atm-00.
- Braden, B., Clark, D. and Shenker, S. (1994a) *Integrated Services in the Internet Architecture: an Overview*, RFC 1633.
- Braden, B. et al. (1994b) *Internet Architecture Extensions for Shared Media*, RFC1620.
- Callon, R. et al. (1997) *A Framework for Multiprotocol Label Switching*. Internet-draft, draft-ietf-mpls-framework-00.txt.
- Comer, D. (1995) *Internetworking with TCP/IP*, Prentice Hall, New Jersey.
- Demers, A et al. (1989) *Analyses and Simulation of a Fair Queueing Algorithm*. Proceedings. ACM SIGCOMM.
- Drury, D. (1996) ATM traffic management and the impact of ATM switch design. *Computer Networks and ISDN Systems* **28**, 471-479.
- Garrett, M. (1996) *Interoperation of Controlled-Load and Guaranteed-Service with ATM*. Internet-draft draft-ietf-issll-atm-mapping-01.txt.
- Georgiadis, L. et al. (1996) Efficient Support of Delay and Rate Guarantees in an Internet. *Proc. ACM SIGCOMM 96*.
- Heinanen, J. (1993) *Multiprotocol Encapsulation over ATM Adaptation Layer 5*. RFC 1483.
- Intserv (1996) From the IETF Integrated Services (intserv) discussion list, after the publishing on the list of the paper, "A study of bandwidth requirements for guaranteed service traffic with WFQ scheduling" by S. Keshav
- ITU-T (1993a) Recommendation X.75, *Packet-Switched Signalling System Between Public Networks Providing Data Transmission Services*.
- ITU-T (1993b) Recommendation X.25, *Interface Between D C E and D T E for Terminals Operating in the Packet Mode on Public Data Networks*.
- ITU-T (1994) Recommendation Q.2931 *B-ISDN User-Network Interface Layer 3 Specification for Basic Call/Bearer Control*.
- ITU-T (1996a) Recommendation Q.2962 *Digital Subscriber Signalling System No. 2-Connection characteristics negotiation during call/connection establishment phase*.
- ITU-T (1996b) Recommendation Q.2963.1, *Digital Subscriber Signalling System No. 2-Connection modification: Peak cell rate modification by the connection owner*.
- Laubach, M. (1994) *Classical IP and ARP over ATM*. RFC 1577.

- Luciani, J. et al. (1997) *NBMA Next Hop Resolution Protocol (NHRP)*. Internet draft draft-ietf-rolc-nhrp-11.txt.
- Malcher Bastos, C.A. (1997) Connectionless service for Guaranteed Traffic in IP internets over ATM (in Portuguese). Doctoral thesis. Dept° de Informática, Catholic U. of Rio de Janeiro, Brazil.
- Malcher Bastos, C.A. and Stanton, M.A. (1998) Efficient Support for Guaranteed Service in IP over ATM, accepted for presentation at 9th IEEE Workshop on LAN/MAN, Banff, Canada.
- Nagle, J. (1987) On Packet Switches with Infinite Storage. *IEEE Trans Com*, **35** No 4.
- Partridge, C. (1994) Gigabit Networking. Addison Wesley Pub. Co.
- Parekh, A. (1992) *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks*, Technical Report LIDS-TR-2089, Laboratory for Information and Decision Systems, Mass. Inst. of Tech.
- Perez, M. et al. (1995) *ATM Signalling support for IP over ATM*. RFC 1755.
- Rampal, S. (1996) *Flow Grouping for Reducing Reservation Requirements for Guaranteed Delay Service*. draft-rampal-flow-delay-service-00.txt,
- Shenker, S. et al. (1997) *Specification of Guaranteed Quality of Service*. Internet-draft draft-ietf-intserv-guaranteed-svc-07.txt.
- Wroclawski, J. (1996) *Specification of the Controlled-Load Network Element Service*. Internet Draft draft-ietf-intserv-ctrl-load-svc-04.txt.
- Zhang, L. et al. (1993) RSVP: A New Resource Reservation Protocol, *IEEE Network*, September 1993.

## 6 BIOGRAPHY

### *Carlos Alberto Malcher Bastos*

Associate Professor in the Telecommunications Engineering department of the Universidade Federal Fluminense, where he has occupied several technical and administrative positions, and where he has been active for several years in data communications and computer networks. He holds a doctoral degree in computer science, and his current research interests are high-speed networks, ATM, integrated services and IP over ATM.

### *Michael Anthony Stanton*

Originally from Manchester, England, gained his B.A and Ph.D. at Cambridge University, before settling in Brazil. Presently holds the positions of Associate Professor at the Catholic University of Rio de Janeiro, and Full Professor at the Universidade Federal Fluminense, both in the field of computing. Current research interests are in computer networking and distributed systems, with emphasis in high-speed networking, network management and security.

# **Part Eight**

---

## **Internet Applications**

# Orchestra ! : an Internet service for distributed musical sessions and collaborative music development and engineering

*P. Bussotti, F. Pirri*

*Dept. Of Electronic Engineering, University of Florence*

*Via di S. Marta, 3 50100 Florence, Italy*

*telephone : 039-55-4796370 fax : 039-55-488883*

*E-mail : bussotti@sunto.ing.unisi.it, fpirri@ing.unifi.it*

## **Abstract**

This paper proposes a platform for musical sessions distributed over the Internet and for music development and engineering. The system is based on a distributed software architecture and it's universally accessible through a simple http-client ( a browser ). Orchestra! is a *service* that offers a common working environment and tools to the users, as well as a *platform* that provides a layer of common functionality for existing musical and groupware tools and for the development of new ones.

The main challenge of Orchestra! is to offer an useful and appealing service with little resources required on the client's side (i.e. a sound card and a browser) while providing an appreciable solution to the typical problems of collaborative real-time applications over the Internet, within the limits of this particular case and its requirements : musical group synchronisation.

As it is well known, packet-switched network's delays make impossible to reach a musical synchronisation among all members of a virtual group when each one is connected to the others. Nevertheless, by conjugating the ideas of hierarchical

rhythmic tree organisation ( from professional musical recording ) and of store-and-forward groupware systems, a stream communication model had been developed which is insensitive to delays. This should give an appreciable approximation of real-life musical sessions, with the remarkable plus-value of allowing distant people to play together, as well as providing a solution to the problem of rooms and equipment for group music.

Orchestra! provides basic functions for group music, such as the possibility to perform together across the net and to inhabit a collaborative environment where collective developing of music and multi-track recording is possible.

These can be taken as a basis for specialised applications in different fields: for example, in musical didactics a particular session arrangement would be required in order to fit the needs of teachers and classrooms, as well as in professional music security requirements could be fundamental, together with the needs for more sophisticated sound processing elements to be added as plug-ins of Orchestra!.

### Keywords

**Group synchronisation, streams, group session, collaborative environment**

## 1 INTRODUCTION

In recent years the Internet has been rapidly evolving into a popular source of information and more and more useful and appealing services, in accord to its underpinning philosophy of a large and low-cost network. In this direction, the creation of the World Wide Web had first contributed to a more efficient information flowing through University Centres and, later, to the creation of a real common cultural heritage. Besides, shareware software has become a common resource in many *anonymous ftp* sites.

A fundamental step of this evolution has been the idea of *distributed software architecture*, a deeply innovative approach that, in its various forms, tends to free the local host from the actual possession of a specific application software by making its functionality available when required within a client-server paradigm.

The wide spreading of software models like Java, ActiveX and Corba clearly confirms this tendency, that's moving further towards even more radical solutions, like the so called *network computers*: they should even abandon the resident operative system for a distributed one. In this sense the application becomes an *Internet service*, the access to which can be left free or regulated by on-line registrations and certifications. The object-oriented or, at a deeper level, the component-oriented technology represents a powerful foundation for distributed architectures. The final user himself takes advantage of it because he

can personalise its application by aggregating those components that best fit his needs.

Those techniques had also proved to be a very effective basis for the developing of *collaborative applications*, which is one main goal of telematics, both in the fields of distance learning and distance working.

Orchestra! gives its contribution in this direction, by developing a multimedia collaborative environment over a distributed software architecture.

## 2 BACKGROUND

Computer music is an example of how art and technology can co-operate to create new expressive forms. Beside the developing and spreading of MIDI keyboards and samplers, popular software like *Cubase* (1998) or *CakeWalk* (1998) had deeply changed the approach to music of groups, both at the professional and at the *amateur's* level.

Group music over the Internet is a natural extension of computer music and it's still a young field of research.

It encounters the critical problem of exchanging over a packet-switched network synchronous streams of data with an extremely low tolerance from delays. In fact, while playing in real-life, everyone gets an immediate feedback from himself and the others, and this permits the group synchronisation.

In other works about this argument, well documented in Paradiso (1997), like *Spinning Disks* of E. Metois or the *Palette of Brain Opera* at the Massachusetts Institute of Technology (MIT), based on the work of J. Yu, the focus has been put more on the creative and artistic aspect of working interactively with expert-system generated sounds rather than on simulating a real-life musical group session (in this context we will relate to *group session* in the meaning of collective activity both in the real-life situation and in the Orchestra! framework).

Orchestra!, on the other hand, deals with audio streams (i.e. the voice, analogic instruments) as well as MIDI, and thus it suffers much more from network and processing latencies. That's why Orchestra! implements a specific strategy of *masking the net to the user at the application level*, by allowing only a cascade-like streaming among the members' hosts with each one mixing his contribution along the cascade, according to the natural levels of rhythmic priority.

Important contributions to Orchestra! design came also from recent literature about *DVEs* (*Distributed Virtual Environments*), especially for object-oriented techniques for manipulating common objects and for distributed system architectures for large numbers of users (Braham, 1997)(Barrus, 1997)(Rockwell, 1997)(Roehle, 1997)(Clark, 1997)(Anderson, 1997).

The lower level problem of intrinsic synchronisation of streamed audio ( the term *intrinsic* is here used to distinguish it from *musical group* synchronisation)

has been approached using *RTP (Real Time Protocol)* technology background (Casney, 1996).

In this paper, we will refer loosely to *connections* in the meaning of point-to-point sessions, independently from the underlying transport protocol.

### 3 USER NEED ANALYSIS AND ORCHESTRA! SERVICES

From the analysis of needs of different typologies of users, it arises immediately that three main user profiles has to be considered, that is:

1. the professional user;
2. the user for didactic aims (teacher and student);
3. the *amateur* user.

The Orchestra! approach is to consider the third one (the *amateur* user) as a sort of *basic user*, in the sense that, if proper services were provided to him within a *flexible object-oriented platform*, then each other profile could be satisfied as well by specialising basic functions into more sophisticated ones and by adding new functions.

For example, the interaction management protocol system is held at a separate level from the actual function implementation. In such a way the *peer-to-peer interaction protocol* that is implemented in an amateurs' musical group, can be replaced by a new module performing a *master-slave interaction protocol* that better fits a music classroom's needs. Another example could be the professional user who can add security policies and more sophisticated signal processing systems as plug-ins of Orchestra!.

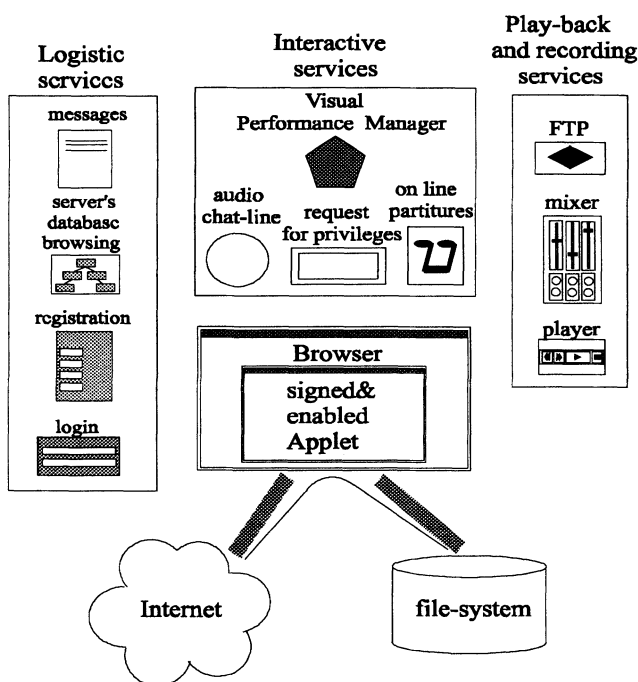
Referring to our target user, the focus has been put on the following points:

1. The application has to be accessible as an Internet service, and it should require the smallest amount of hardware resources on the client's side ( in this case : a bi-directional sound-card and an interface to analogic/MIDI musical instruments or a microphone).
2. There has to be a high level of collaboration : every decision about a *common resource* has to be taken collectively after a discussion.  
For this aim an *audio chat-line* has been provided, connecting all the member of the group among them. The access to the common resources is regulated by a *privilege management system*, and a *privilege access protocol* is implemented in order to obstruct undemocratic actions.
3. It must ensure a good quality of musical performance among people distributed over the Internet. That means that a solution (at least partial ) has to be found to the problem of musical synchronisation in presence of network's delay.



4. Tools have to be provided for multi-track recording and sound engineering.
5. *Music development* is one of the most important and stimulating activity of a group, the one in which each individual brings his own ideas and confronts with the others in a constructive way. Therefore, adequate tools have to be provided to allow such an activity to be carried out within a net session.
6. All the tools have to be available also outside the collective session in such a way that everyone can work and practice individually with musical bases and tracks stored in his own hard disk.
7. External people could be allowed by the group to participate as an *audience* at the group's live performances.

Figure 1 illustrates a summarising scheme of Orchestra! services (user interface), classified as *Logistic services*, *Interactive services* and *Play-back and recording services* (the scheme refers to our implementation using Java™ Applets, but it can be implemented in several other ways).



**Figure 1 - Orchestra! services**

A more complete description of the Orchestra! platform will be given in the following sections :

- the system architecture;
- the hierarchical rhythmic tree approach;
- the collaborative environment;
- our prototype's implementation.

### 3.1 The system architecture

Orchestra! is based on a distributed software architecture that allows the application module to be accessed by multiple users from a common *module server* through a simple resident client application. This module is put in execution on the local machine and every host receives an identical copy of it : in this sense it is an *Internet service*, which frees the local host from actual possession of software, with economic advantages both for what concerns acquiring it and for memory occupation on the local hard disk. In particular, such an application will not suffer from *versioning* and the necessity of periodical software upgrading.

Orchestra! establishes a net connection among the dislocated hosts which belong to a common musical session, in order to implement the sharing of audio streams, musical tools and session data within a collaborative environment.

For this reasons, each single Orchestra! module has to be able to open connections towards other hosts, as well as to access local resources, as the file-system for storing works, or for loading local libraries.

These capabilities are strongly dependent on the particular software architecture; for example, Java™ Applets require a special signing and privilege-enabling procedure to access resources outside its *security sandbox*, while ActiveX Controls don't suffer from any security restriction.

In order to implement an effective *session management*, Orchestra! uses a client-server paradigm at the application level by pairing an "*Orchestra! Session Server*" to the "*Orchestra! Module Server*" ( that is a Web Server in our Java implementation). The *Orchestra! module* is, then, the client side of the system : it creates a connection towards its *session server* as soon as it starts to run on the local machine.

The *Orchestra! Session Server* manages the sessions in a centralised way, by :

- keeping an updated database of groups and individuals registered to Orchestra!;
- keeping the current *status* (e.g. session on/off, play/idle, list of participants etc.) of each session;
- storing common group resources (as musical tracks, masters etc.);

- allowing the access to common group resources in a collaborative way and regulating it by a *privilege access protocol*;
- keeping the table of *hierarchical tree* connections ( see paragraph 3.2 - the hierarchical tree approach) and allowing a collaborative modification of it through the *Visual Performance Manager* on the graphical user interface;
- notifying every change occurred in the session status to all the participants;
- providing a message mailbox for internal use of a group;
- *tunneling* and broadcasting audio streams.

Every action of the individual member on the system is broadcast to all the others. In particular, the *audio chat-line* uses the server to collect audio-streams from each one, mix them and then broadcast them to all the others.

In our implementation the server manages all the connections, acting as a ***broadcaster*** for *chat-line* audio streams and as a ***tunnel*** for the *guide-stream* of the *Performance system* ( see paragraph: 3.2 - the hierarchical tree approach). This choice is in complete agreement with the idea of giving to the server the total control over the session, in such a way that the list of current participants and the related configuration of connections is made *transparent to the user*.

Orchestra! manages the connections' system at the Application Level of the OSI-Reference Model . It must be remarked, however, that multicast techniques could offer effective solutions, as well. When dealing with a large number of users the best solution could be a hybrid system with several servers communicating together.

### 3.2 the *hierarchical rhythmic tree* approach

As it is well known, the problem of *delay*, that's inherent to the nature of a packet-switched, non-homogeneous internetworking, gives a limit to the responsiveness of interactions, and thus to the quality of real-time collaborative environments. In order to focus on the real *core* functionality it has been decided not to deal with video data but only with *audio, image and textual communication*, which helps decreasing drastically the amount of required bandwidth. As it can be easily argued the critical point for an application that tries to reproduce a real-time interactive musical session is that it should reproduce the quasi-instantaneous propagation of sound within a small room: that's, in fact, the necessary condition for instrument players and vocalists to synchronise with each other. Of course, this is *not* achievable over the Internet, and the musical track of a player B that receives a track from a player A and synchronises himself on it will return to the player A after twice the connection delay, that is unacceptable. Nevertheless, this work would try to show how even a very critical interactive situation (perhaps the most critical) could be approximated up to a certain degree of acceptance by a network application if proper strategies are adopted.

The key idea comes from the techniques of professional musical recording , and, in particular, from the *hierarchical rhythmic tree* approach, where musical tracks are recorded in a sequence that starts from the instrument that has the highest *rhythmic priority* ( i.e. the drums) down to lowest rhythmic priorities (i.e. bass guitar, rhythmic guitar, keyboards, soloists, vocalists).

In a professional recording context, this is used for ensuring a stable rhythmic basis for melodic instruments and voices that, for their congenital extremely variable performance (i.e. melody) , are more subject to losing synchronisation.

In our context, we use this priority ordering to define a system of network connections among players : the drummer will play alone and communicate its *captured* musical stream to the bass guitar player who will synchronise himself on it and mix its contribution to the stream : it's evident that this can be done no matter how long the delay is. Then, in turns, he will communicate the *digitally mixed* stream (in which drums and bass are synchronised) to the rhythmic guitar player and so on. The lowest steps of the cascade (i.e. soloist, vocalist) will receive a stream that's acceptably rich of previous instruments' contributes, and this should give a sensation that's very close to the real-life session.

The constraint of this cascade system (which can be easily expanded into a tree structure) is that everyone receives the stream that comes from higher priority instruments and delivers it to lower priority ones : the fact that none gets a stream from someone he had previously sent a stream to, ensures that network latencies won't affect the correct functioning.

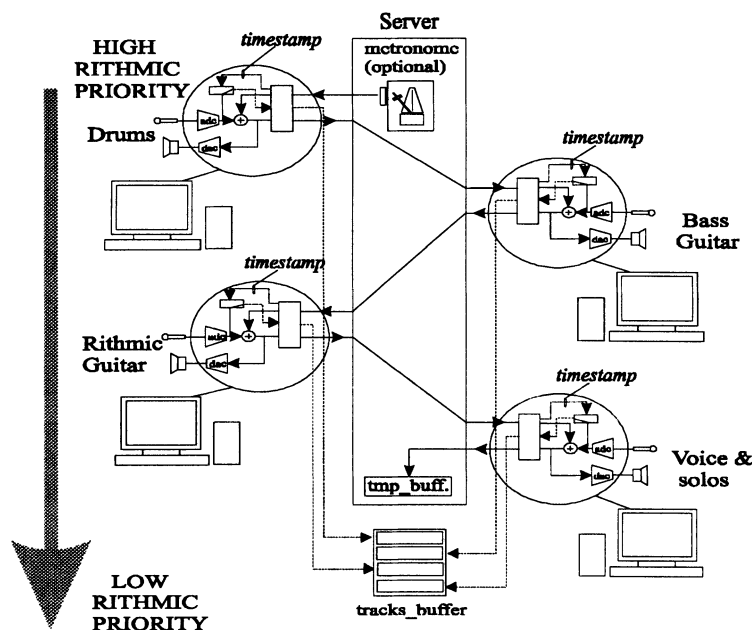
This uni-directional audio stream that passes through the cascade has the only function of allowing a right synchronisation of all the instruments and of creating an effective *ensemble* feeling ; this "*guide stream*" will occupy just one channel, because each new contribution is digitally mixed to the incoming stream.

Meanwhile, each player produces also a *copy* of his own track that's kept separate from the mixed stream; this is marked in a proper way (by using RTP timestamping), in order to save time references for final tracks' mixing, and sent to a server's *tracks' archive*. It will keep a vectorial representation of each song in terms of separate tracks, which can be singularly edited or overwritten in a second time by an authorised person (e.g. each player can be authorised to perform I/O with his own tracks and read-only access to others' tracks).

At the end of the cascade the *guide stream* can be discarded or broadcast to the audience : in each case, the final high-quality product is the master resulting from tracks' mixing.

A *tracks' mixer* is available, and the session member who have received a proper *Mixer Operator token* from the others is allowed to access all the tracks and mix them into a *master*, that is successively stored in the server space of the group as a read-only file.

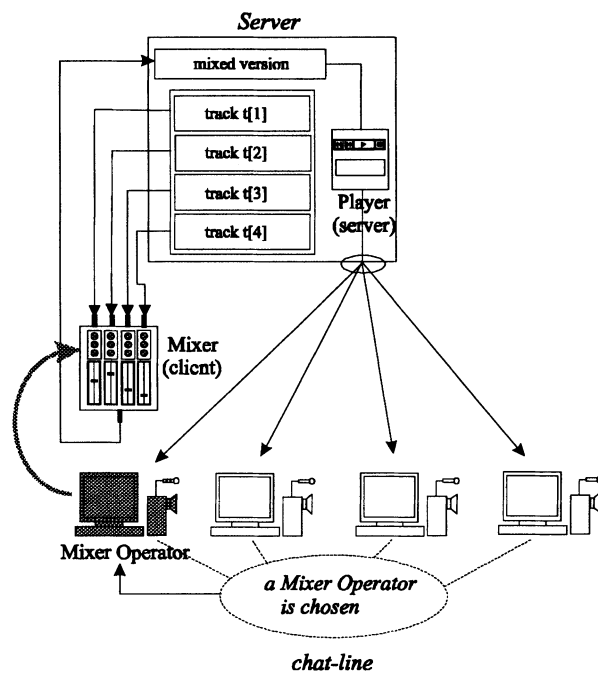
Figure 2 illustrates the basic guide-stream flowing through the cascade of hosts. Each of them receives it from the immediately upper priority instrument, adds it to its instrument's captured audio stream and sends the resultant stream to lower priority instruments. As far as the internal delay of the audio *capture* and *playing* system is a deterministic and known value, a synchronisation of the two stream is easily achieved (for sake of clearness in fig.1 the internal delay is considered to be null). At the same time, each instrument produces a copy of the track and stores it separately in the server's track-buffers. Of course, each track could be stored in the local hard disk and sent to the server at a second time, as well.



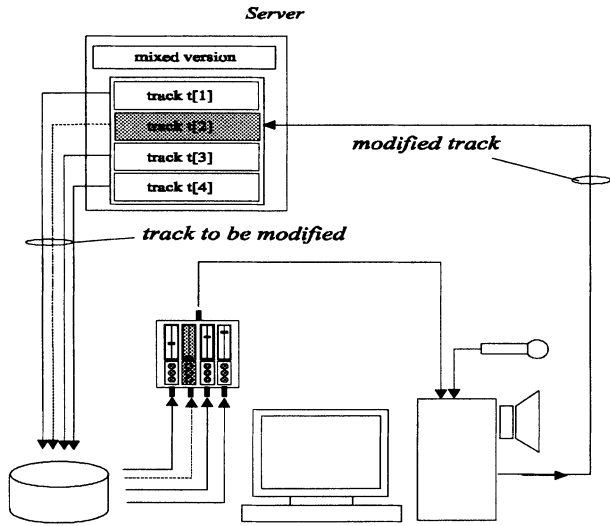
**Figure 2 - guide-stream flowing through the Orchestra! cascade (tree)**

It's convenient to use a low bit-rate *waveform* stream for the *guide-stream*, in order not to overload the network in time-sensitive operations, while the *track* should be a high-quality one or, when possible, a *MIDI* track (in this case the mixer has to implement MIDI format).

It could be argued that a system like this could work as well with a real *store&forward* approach, where the *whole block of waveform data* is passed as a file through the cascade and, at each step a new instrument is added (this won't require anything more than FTP). This is true, but the choice of *streaming real-time captured audio* allows a much more satisfying *ensemble* sensation, for example when using the chat-line for interrupting the performance at an arbitrary point.



**Figure 3 - tracks' mixing and broadcasting of the master.**



**Figure 4 - Modifying of a track (stand-alone mode)**

Figure 3 illustrates the mixing phase and the playing of the master.

Figure 4 illustrates the single player that, *out of the session ( or in a stand-alone mode )*, substitutes his own track with a new version.

To do this, he can access the session server to get all the song's tracks and store them locally, mix all the track except his own and then use this incomplete mixed version as the basis to play along with.

Then, he can overwrite his own track on the server and leave a *message to the group* for advising all the others about the replacing of his track.

The Orchestra! sub-system that manages the structure of the *hierarchical rhythmic tree* is named *Performance System*. It has a default way to organise currently present instruments in a tree structure, according to their rhythmic priority level and to the current list of session participants; in particular, if an instrument enters a session that's already begun, it'll be automatically inserted at its right place.

It must be noticed that, for the already mentioned reasons, if a node contains two or more parallel instruments (i.e. with the same priority) the system must decide which way the guide-stream will follow (two parallel streams can't be added together : one must be discarded), and this is done on a default basis which can be modified.

The Performance System plays also a key role in managing the *external access to a session as Audience* ( see paragraph 3, point 7 : User need analysis and Orchestra! services).

In fact, all the people who obtain an access to the session with an *audience privilege* is set by the Performance System at the end of the cascade ( or of the main branch of the tree) in such a way that they can listen to the most complete version of the guide-stream.

Besides, the Performance System offers to the group a graphical interface that allows a visual editing of connections and a runtime monitoring of connections' status: the *Visual Performance Manager*.

Using this tool, after having received a proper *Performance Manager token* from the rest of the group, the default rhythmic tree can be modified in an arbitrary way, and this is particularly useful in cases like these :

- equal priority instruments can be arranged as consecutive;
- consecutive instruments can be arranged at the same hierarchic level;
- there can be set a different path for the guide-stream in cases of nodes with two or more instruments ( i.e. proper *tree* structure instead of a *cascade*).

Another fundamental service which can be easily handled by the *Visual Performance Manager* is the *music developing environment*.

In fact situations like the following :

- one player presents his idea to the others;

- two instruments ( or, in general, less than the total number ) practice to play together while the others are listening,

can be managed entering the *Developing* mode of the Visual Performance Manager. In this mode (as opposed to the *Playing* mode) not all instruments are assumed to be playing, and the *active* ones must be indicated by marking their icons on the graphical interface. All the unmarked instruments will be temporarily put at the end of the cascade, as listeners.

An evident problem of this system is that the *ensemble* feeling could be reasonably get by the only low priority instruments, which get a rich and articulate guide-stream from the upper rhythmic instruments. This could be obviate when the song is strictly synchronised by a metronome : in these cases a *local guide-stream* , containing only some lower tracks previously recorded, could be locally played with the drums or with the bass and then stopped inside them. This won't compromise anything if a correct metronome timing is used (notice that, in absence of a metronome, tempo and rhythmic variation can be decided by the drummer according to his improvisation ).

### 3.3 The collaborative environment

Orchestra! has been conceived as a collaborative environment where people could use telematic resources to perform all activities related to music in a realistic way. In particular, music is a field where the *emotional factor* is predominant over many other aspects and this has to be well analysed in order to meet a music player's expectations. Musical emotion arises from a mixture of *acoustic satisfaction* ( that's why a high sound quality is required) and *ensemble feeling* (that 's why everyone have to feel in close contact with the others).

From personal experience in musical groups, I think that the pairing of an effective *Performance System* (with *Playing/Developing* modes) with an *audio chat-line* could be more important to a player than, for example, video data : *music can be enjoyed with closed eyes*.

The *chat-line* should be the main way to develop ideas, express comments and decide about common interests : that is, the real-life situation of the group that meet in a private music room has to be simulated as much as possible. Of course, the problem of network's delay will afflict the chat-line as well, but under certain limits, this will not compromise the discussion. In each case, the *unregulated chat-line* could be replaced, when required, by a *regulated chat-line*, where some policy for sharing the audio channel is implemented.

Orchestra! defines *common resources* of a group like the current and old tracks, recording masters and so on, and offers to the user several tool to work with them in a really collaborative way . At the same time, the Performance



System offers a Visual Interface for modifying its default configuration in a simple and intuitive way.

### 3.4 Our prototype's implementation

Our prototype of Orchestra! uses Java™ Applet technology (JavaAPI 1.0) for the implementation of the application module. In this case the *module server* is the web-server and the client application for getting it is a http-browser. This makes Orchestra! platform-independent. We opted for a Java implementation of Orchestra! mainly for the high level of integration of Java within the World Wide Web and for its independence from platforms : the choice of CORBA would have required a specific software installation on the client's side, while DCOM techniques (ActiveX) would have limited Orchestra! to Microsoft platforms.

We are creating an Orchestra! web-site where Orchestra! service is available after a registration phase. As far as JavaAPI 1.x do not provide any class for handling audio-streams (capturing, playing) *native C methods* had been interfaced with the Java code in order to perform the mentioned tasks. This requires a dynamically linked library to be installed on the local machine and this is done at the first access to Orchestra! site by anonymous ftp.

Independence from platform is maintained by providing proper audio libraries for the most common operative systems. The access to local resources, like read/write on the file system or loading a resident library could have been implemented by the use of *Netscape™ Capabilities Classes* : this approach makes each local resource available to the Applet which is properly signed and authorised. On the other hand, this limits the choice of browsers to Netscape 4.0 or more recent versions.

## 4 CONCLUSIONS

As a conclusion, it must be stresses again the fact that Orchestra! is fundamentally a *platform* , a container that's flexible enough to support every different specialisation of its basic functionality, and to embed new components in its collaborative environment.

The main direction of our research will be, therefore:

- enhancement of Orchestra! performances;
- flexibility towards new audio formats;
- extensions of Orchestra in the directions of Music Distance Learning and Professional Music.

Another way we are expanding Orchestra! concerns the creation of a DVE where people can move along rooms and corridors : each active musical group will be assigned a room, and the visitor will listen to their music as soon as he enters that

room. This could be a nice way to look for musical contacts with already formed groups in Orchestra!

## 5 REFERENCES

Cubase (1998) , trademark of Steinberg Vertrieb GmbH.  
homepage : [www.steinberg.de](http://www.steinberg.de)

CakeWalk (1998), trademark of Twelve Tone System, Inc.  
homepage : [www.cakewalk.com](http://www.cakewalk.com)

Paradiso, A.J. (1997) Electronic music: new ways to play. IEEE Spectrum, December 1997, volume 34, number 12, 18-30.

Braham, R. and Comerford, R. (1997) Sharing virtual worlds. IEEE Spectrum, March 1997, volume 34, number 3, 18-19.

Barrus, J.W. and Waters, R.C. (1997) The rise of shared virtual environments. IEEE Spectrum, March 1997, volume 34, number 3, 20-25.

Rockwell, R. (1997) An infrastructure for social software. IEEE Spectrum, March 1997, volume 34, number 3, 26-31.

Roehle, B. (1997) Channeling the data flood. IEEE Spectrum, March 1997, volume 34, number 3, 32-38.

Clark, R.K. , Franceschini, R. and Reece, D. (1997) Synthetic soldiers. IEEE Spectrum, March 1997, volume 34, number 3, 39-45.

Anderson, D.B. and Casey, M.A. (1997) The sound dimension. IEEE Spectrum, March 1997, volume 34, number 3, 46-50.

Casner, S. , Frederick, R. , Jacobson, V. . Shulzrinne, H. (1996) RTP: A Transport Protocol for Real Time Applications. RFC 1889, January 1996.

## 6 BIOGRAPHY

**Paolo Bussotti** was born in Florence on July 20<sup>th</sup>, 1966. He received the Laurea (Dr. degree) from the University of Florence in 1996. He is currently a Doctorate student (II year) in Telematics at the University of Florence, where he is working on distributed software applications and systems, especially on the Java platform .

He's also working on image and video processing and computer vision.

**Franco Pirri** was born in Livorno on May 14th, 1945. He received the Laurea (Dr. degree) from the University of Pisa in 1971. He joined the Dipartimento di Ingegneria Elettronica of the University of Firenze in 1973 and became Associate Professor of "Industrial Electronics" in 1982. He is currently Associate Professor of "Telematics" and of "Automatic Design of Electronic Circuits and Systems".

Franco Pirri is author of more than 75 papers, congress presentations and technical reports. His professional and academic experience has been focused on digital system design, with emphasis on communication networks, multimedia applications and application specific integrated circuits (ASIC).

He is member of the IEEE, the IEEE Computer Society and the IEEE Communication Society.

# High-Performance Online Presentation of Complex 3D Scenes

*S. Olbrich, H. Pralle*

*Lehrgebiet Rechnernetze und Verteilte Systeme (RVS),  
Universität Hannover*

*Schloßwender Str. 5, D-30159 Hannover, Germany*

*Tel.: +49 511 762 3078, Fax: +49 511 762 3003*

*email: olbrich@rvs.uni-hannover.de*

## **Abstract**

Online presentation of virtual 3D objects in the Web, based on Internet standards, is limited due to several performance bottlenecks, quality restrictions and missing functionality. Particularly in the scientific context, where high-performance client, server, and network equipment exists, and requirements for high complexity of represented 3D geometry are given – e. g. in the case of scientific visualization of large datasets – the potential performance of such scenario is not utilized. This paper describes the concept, implementation and evaluation of an optimized viewer, based on a new 3D stream format. The design of this 3D representation and the strategies realized in the viewer were tuned for efficiency, especially to take advantage of high bitrates to obtain short latency. This led to the feasibility of streaming sequences of 3D objects, applying the „Real Time Streaming Protocol“ (RTSP) to enable on-the-fly presentation as a 3D movie, freely navigatable at the client side, using virtual reality methods, such as stereoscopic presentation in conjunction with tracking systems.

## **Keywords**

Virtual Reality, VRML, Scientific Visualization, Hypermedia, Browser, Plugin.

## **1 INTRODUCTION**

Distributed multimedia information services in the actual discussion are based on protocols, addressing schemes, and services that are established in the Internet. These are described in Internet Standards called RFCs (Request for Comments),

e. g.: TCP/IP, URL [3], HTTP [5], MIME [6], HTML [2]. The combination of these Internet techniques – well-known as WWW (World Wide Web) – allows the development of interactive, distributed applications that take advantage of the integration of different media types: text, hypertext, image, graphics, video, audio, and 3D objects – or compositions of these, timely or spatially synchronized, as hypermedia documents. On the client side, a generic browser – such as Netscape – serves as the user interface with extensible presentation and interaction capabilities.

3D technology is useful in different application areas of information systems to take advantage of three-dimensional, ergonomic user interfaces, adapted to virtual reality metaphors, for example:

- as a method to navigate in an information space, or
- for online presentation of virtual 3D scenes, e. g.:
  - reproductions of real objects or
  - artificial scenes, such as results from scientific visualization.

In the application of 3D techniques that are established in the WWW, such as VRML (Virtual Reality Modeling Language) – applied in Version 1.0 [1] and 2.0, now called VRML97 [9] – in conjunction with appropriate viewers (see also [19]), several limits have been observed. For certain scenarios, especially in the context of

- high quality application requirements, e. g. handling objects with high complexity,
- network infrastructure offering high bitrates, such as local networks,
- high performance server and client systems,

which exist in scientific and industrial research environments, several constraints regarding performance, quality, and functionality aspects prohibit any useful application, in particular:

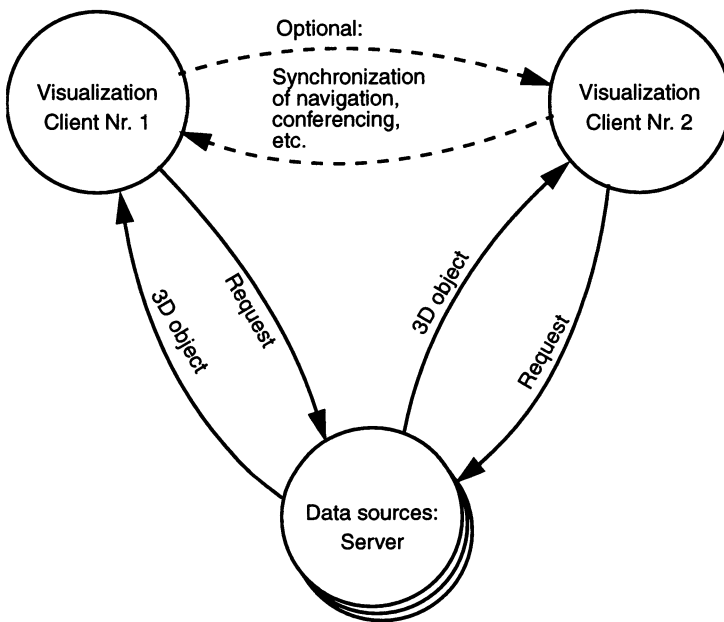
- unacceptable delays from request to presentation,
- no progressive presentation capability,
- low frame rates while navigating in the scene,
- low quality of rendering – e. g. no support of antialiasing,
- little support of immersive virtual reality methods, such as stereo presentation and tracking systems.

In the following, the requirements in our focussed application scenario and reasons for the disadvantageous characteristics of current implementations are analysed, resulting in the introduction of an innovative approach and the evaluation of a prototypic realization for the accelerated online-presentation of virtual 3D objects: *DocShow-VR*.

## 2 SCENARIO: „SCIENTIFIC VISUALIZATION“

We focus on applications of 3D information services to support visualization applications, in order to contribute advanced methods for high-quality online presentation of scientific results. The typical considered working environment in science and research consists of

- high-performance graphics workstations – „Clients“, and
- data sources, such as compute or information servers – „Servers“,
- connected to high-speed local and/or wide area network – „Intranet/Internet“.



**Figure 1** Visualization clients in a collaborative, distributed system.

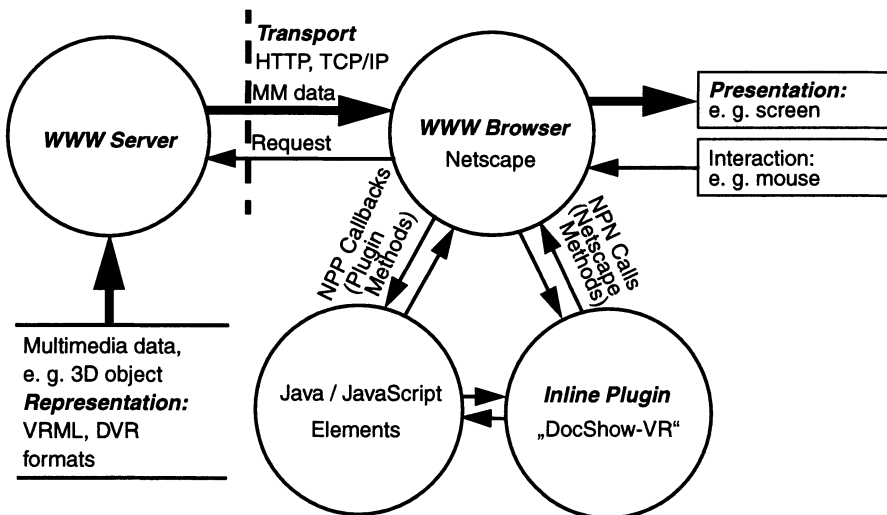
In the scientific visualization context, high-quality services and applications are required because of the constraint not to allow lossy processing steps, that could result in misinterpretation of the representation. Involved are several data and media types, in particular vector-oriented 3D objects. Compression algorithms – if considered at all – have to be lossless in order to preserve the accuracy of the representation. Besides that, they have to be implemented efficiently in software, since hardware solutions for such special problems are not realistic for the small number of products on the market.

This is in contrast to consumer applications, where lossy compression techniques – such as JPEG and MPEG for 2D raster images and videos, respectively – are widely used.

For data exploration of high-volume, multidimensional simulation or measurement results, interactive, three-dimensional computer graphics systems are appropriate, and immersive, virtual reality systems are increasingly required – often supported by specialized visualization software. To take advantage of the underlying 3D representation for platform-independent, navigatable high-quality online presentation, the requirements are:

- standardized 3D file format – such as VRML (Virtual Reality Modeling Language) [1][9][19],
- open export interfaces for 3D file format in the visualization software – e. g. „write VRML“ module for AVS [22],
- generic WWW browser – such as Netscape,
- 3D viewer as external application or as inline-plugin – e. g. Cosmoplayer,
- presentation and interaction devices.

With this prerequisites it is, in principal, possible to construct a system for publishing visualization results by storing VRML files on a WWW server and accessing and presenting them via Internet protocols, as shown in Figure 2.



**Figure 2** Multimedia online publishing in the World Wide Web: client/server model.

In Figure 2, the viewer is integrated as an *Inline-Plugin*, which is realized as a shared library, according to the conventions specified by Netscape [15]. If a URL is called whose content matches the MIME type that is supported by this plugin, a presentation window is opened inside the Navigator or Communicator application, and the data is streamed directly into the plugin via callback routines provided by the plugin software. As such, this interface works significantly more efficient than the external viewer application call mechanism using a local file copy, in particular allowing smaller latency and higher throughput.

An Inline-Plugin can also be called by using an EMBED-Tag, similar to the IMG-Tag for presenting GIF or JPEG images, embedding such a presentation in an HTML page. Besides that layout-supporting capability, Netscape has defined a programming interface that allows to communicate with Java and JavaScript – this is called *LiveConnect*. By applying this mechanism it is, for example, possible to offer plugin functions that can be controlled by user-created buttons on the same HTML page.

### 3 PROBLEMS

It has been shown that this VRML-based configuration, consisting of available internet tools, is insufficient in the considered scenario, regarding several aspects:

1. Performance,
2. Quality of Presentation,
3. Functionality.

#### 3.1 Performance

The **load times** for complex objects with polygon count in the order of 100.000 or more – which is typical for scientific visualization results – are in the order of minutes. These startup times are much too long for the intended productive working environment, and the interactivity is very restricted. It is mainly caused by processing expense at the client side – available communication networks, such as local networks, allow much higher bitrates. The preparation of the cleartext-encoded VRML format, mostly gzip-compressed, into the binary representation that is suitable for graphics rendering, is very computational expensive. Involved are decompression, decoding, and parsing at the client side.

The **navigation speed** is very slow, often resulting in several seconds latency – where the rendering performance of the considered graphics workstations would allow interactive frame-rates. Possible reasons are:

- Special rendering primitives – such as „triangle strips“ – that are frequently generated by visualization tools and that are optimized for high graphics throughput, are not supported in VRML. The universal VRML polygon lists could on principal be optimized at the client side by automatic recognition of



*independent triangles* and *triangle-strips* and appropriate conversion. But this would be computationally expensive, and the startup time would increase.

- The traversing and rendering steps of the previously decoded and parsed 3D object structure are implemented inefficiently. It should be taken into account that frequent changes in the graphics state and insufficiently designed loops can significantly contribute to inefficiency.

### 3.2 Quality of Presentation

**Aliasing artifacts** are observed on low-resolution displays as „staircase-stepping“. These could be reduced by antialiasing techniques, which are partially available in hardware, without performance degradation, such as multisampling antialiasing on SGI high-end workstations – but this is not supported by currently existing VRML viewers.

A further important issue is to achieve high image quality, that means to **reproduce correct colours**, independently on the presentation device. This could be done by integration of a colour management system in conjunction with object colour specification using device-independent colour space and colour profiles [8], respectively.

Missing capabilities of an immersive virtual reality system:

- **Stereoscopic presentation** is not supported. This is a partially offered mode of 3D graphics adapters, e. g. all SGI workstations are prepared to drive the Crystal Eyes LCD shutter glasses in order to „stereo-view“ interlaced presented stereo images.
- **Three-dimensional** input devices, such as 6-degree-of-freedom head-tracking or spaceball devices, are not supported.

### 3.3 Functionality

**On-the-fly presentation** of 3D objects – that means: incrementally loading and progressively rendering – is not supported. This is caused by the structure of the VRML file format, in which object coordinates, attributes (such as normal vectors and colours), and topology data (as indexes, referring to the first data elements) are separated. In principal, the presentation can begin only after all coordinates and attributes are transferred. In practise, the latency is further increased, because the rendering typically starts after completely loading and parsing the file, by traversing the represented object structure.

Streaming of **dynamic scenes** – that means: sequences of completely different geometries, presented in real-time, as a 3D-film – is not supported. This limitation is particularly caused by missing real-time capabilities of the browsers and the absence of a 3D streaming protocol.

#### 4 SOLUTION: 3D FILE FORMAT „DVR“, INLINE-PLUGIN „DOCSHOW-VR“

To overcome the efficiency, quality, and functionality problems described in the previous section, we have developed a new 3D file format (DVR), an optimized viewer (DocShow-VR), and a VRML-to-DVR converter (wrl1toDVR).

Several innovative approaches are implemented by now:

##### 1. Minimal startup time and „on-the-fly“ rendering, progressively while transfer.

- Decoding and providing graphics data for rendering with minimal computational cost.
  - The DVR format uses the IEEE format for binary representation of float and integer values in network-byte-ordering, similar to the binary encoding of CGM (Computer Graphics Metafile, ISO 8632). This corresponds to the internal representation of most workstations, where these values can be used directly as arguments for the graphics rendering, which is based on OpenGL [13], the high performance, low-level 3D graphics API with the currently widest platform support. Excepted are workstations and PCs based on Intel and DEC CPUs, where one conversion step has to be executed after the transfer. This is done by the macro *ntohl()*, which is applied on 32-bit integer and float values. DVR records consisting of such byte-order-sensitive data are recognized by a flag in the appropriate record header, which is reset after conversion, in order to detect pending conversions, too.
  - For parts of 3D objects where no explicit normal vectors are specified, a conversion process computes the required normals and stores them in the DVR file.
- Storage and transmission of direct coordinate and attribute values, instead of indexing them in a topology section. This allows starting of rendering as soon as possible, and allowing progressive rendering and efficient pipelining of the transport and rendering processes.
- To take advantage of this short-latency, incremental, streaming process strategy, the viewer was implemented as a Netscape inline-plugin, using immediate-mode graphics instead of display lists in OpenGL.
- Usage of network infrastructure providing high bitrates, such as IP over ATM (155 Mbps) in our institute.
- Compression techniques to reduce volume of transmitted data are not yet involved. This is an area for future development, since geometry-based, lossless compression is available [21], and lossy techniques could be used for integration of different levels of detail in the representation [11]. But in the moment we explicitly avoided any computational overhead at the client side that could introduce additional latency.

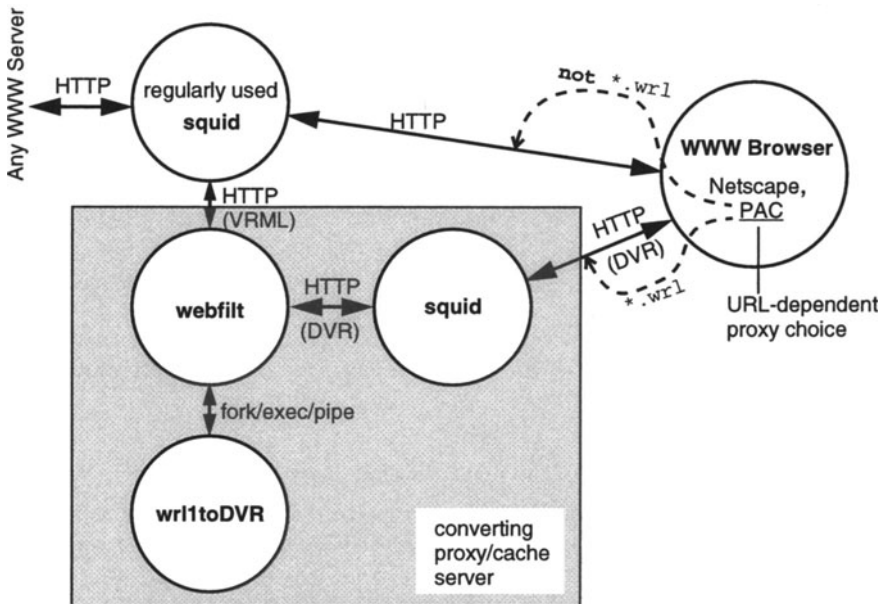
2. Efficient OpenGL rendering support: application of *triangle-strip* and *triangle* primitives besides the universal *polygon* primitives, optimization of graphics status changes, loop optimizations.

- In a preprocessing step, the independent polygons of the VRML format are analysed and consecutive polygons compared, in order to recognize more efficient rendering primitives which are then appropriately written into the DVR file.
- VRML-based information regarding the graphics state, such as material properties or transformation matrices, are converted to DVR records which are output only when necessary – for example, in case of a status change before a new graphics primitive.
- To reduce a CPU bottleneck that could be caused by compute-intensive, frequently executed `if`-statements in sensitive loop kernels (while rendering of attributed sequences of graphics primitives), the possible cases were classified, and for each class separate, optimized routines were implemented. The number of the detected class is coded in the respective DVR record header, so the rendering process can directly execute the appropriate routine. These routines are individually optimized, e. g. by loop unrolling and optional usage of optimized rendering calls, such as `glVertexArray()` in OpenGL 1.1.

3. Compatibility to standards and transparent application.

- A converter has been implemented, providing the preprocessing features as explained above, that takes VRML 1.0 files as input and writes DVR files. At the beginning of our work, VRML 2.0 was under development, but we started to support VRML 1.0, based on the publicly available VRML 1.0 parser library from SGI. In this way we were operating on top of an internet standard for the representation of 3D scenes.  
Our acceleration mechanism is on principal similarly applicable on VRML 2.0, which became an ISO standard (VRML97 [9]), and we will eventually upgrade in this way. But question is if our results would influence the specification of an optional binary coding of a future, revised VRML standard [10] or a virtual reality transport protocol [4].
- Our VRML-to-DVR converter – `wrlltoDVR` – was built into a proxy cache, based on `squid` and a universal `webfilt` tool, which is able to analyse and optionally modify HTTP protocol elements [23]. In a prototypic configuration we provided a Netscape specific PAC (proxy automatic configuration) file in order to access our converting proxy cache for VRML files requested by the user. In case of a cache miss, the proxy automatically converts them using `wrlltoDVR`, called by `webfilt`, and delivers the resulting DVR format to the client, while storing it in the `squid` cache (see Figure 3).

In this way a transparent VRML access and acceleration mechanism was demonstrated. Further developments in this direction could also be suitable for automatic creation of derived media types, such as images or videos or general application for converting services.



**Figure 3** Proxy/cache configuration consisting of squid, webfilt, and wr1toDVR.

#### 4. High-quality presentation and interaction, contributing to immersive virtual reality.

- Aliasing artefacts, such as stair-case stepping on low-resolution displays, can be reduced by wellknown antialiasing techniques. Our viewer supports a hardware-accelerated method called *multisampling antialiasing*, which is offered on SGI high-end graphics workstations as an OpenGL extension, without significant performance degradation.
- Stereoscopic viewing, available on all SGI workstations, is supported by our plugin. The viewing transformation is controlled by the specifications of the PerspectiveCamera node in the original VRML format. The focal-Distance now gets particular significance for the appropriate observer-to-display distance, but the heightAngle must be overridden according to the height of the display and the distance of the observer to the display. The actual constellation has to be calibrated by measuring the width and height of the display, the distance of the observer to it and the eye distance of the

observer and writing these values into a configuration file, which is read by the plugin software at startup. This mechanism has been successfully applied to monitor and large-screen stereo projection devices, using active LCD shutter technique and passive polarizing glasses, respectively.

- Head-tracking systems serve to measure the current position and orientation of the observer. These values are used to control the viewing transformation in order to get a holography-similar presentation. By now, our plugin supports one such device, the ultrasound-based Logitech/Stereographics CE-VR which is integrated in LCD shutter glasses.

## 5. Streaming sequences of 3D scenes.

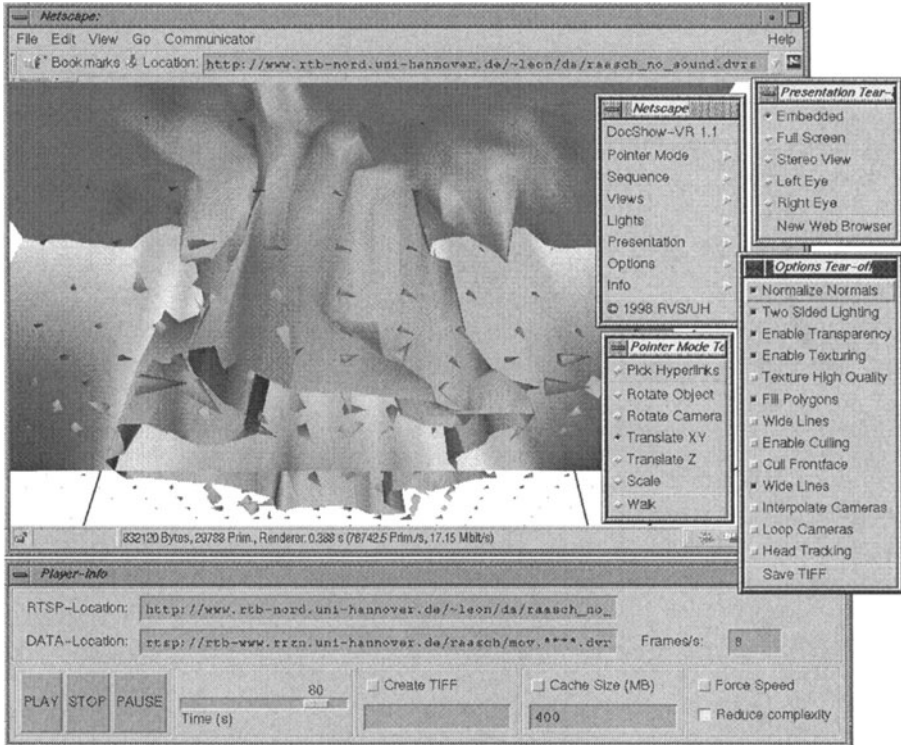
- In analogy to techniques for streaming video and audio media, we have thought about such a mechanism to present time-dependent 3D objects, as „stereoscopic movies“, viewable and navigatable similar to holography. In order to keep synchronization requirements, we can take advantage of
  - our efficient transmission and presentation capabilities,
  - scaling in quality (fine or coarse representation with appropriate level of complexity) and in time (variable time steps), allowing to vary bitrates or rendering rates,
  - and a control protocol already developed for video and audio.

We have extended our plugin to support the *Real Time Streaming Protocol* (RTSP [18]), in conjunction with a special streaming server that delivers *3D streams*. These consist of sequences of 3D scene descriptions in DVR format, terminated by appropriately inserted „end-of-scene records“. Since in this streaming case we had to apply and implement a protocol that is not supported in the Netscape browser, we are requesting first a description file via HTTP, representing a special MIME type that is detected by our DocShow-VR plugin as a script. This is then interpreted in order to request one – or more, then presented as composition – 3D streams. The protocol handling is realized as a separate thread, so that the – eventually moving – scene can be asynchronously navigated, e. g. using mouse-control or head-tracking devices.

The implementation of the VRML-to-DVR converter and the viewer was based on the freely available C++ library for parsing VRML 1.0 from SGI – QvLib [20], an email message from Jan Hardenberg, consisting of fragments of an implementation of a rudimentary, OpenGL-based VRML-1.0 viewer [7], and the Netscape Navigator Plug-In Software Development Kit [14]. The current release of the converter runs on several UNIX workstations – additional converting support is given by a web page, working with a form-based file upload [12], and the DocShow-VR plugin viewer software actually supports UNIX (HP/UX, SGI Irix, SUN Solaris) and Windows 95/NT. For UNIX platforms not providing an OpenGL runtime environment, plugin versions linked with the OpenGL emulation library Mesa [17] were created.

The binaries, the web-based VRML-to-DVR converter service, and several application examples are publicly available:

<http://www.dfn-expo.de/Technologie/DocShow-VR/>



**Figure 4** Application example: „Atmospheric convection“, also showing some popup menus of „DocShow-VR“. Data by courtesy of the Institute for Meteorology, University Hannover.

## 5 EVALUATION: „SGI COSMOPLAYER“ VERSUS „DOCSHOW-VR“

In order to evaluate our implementation, we compared the data volumes, startup times, and rendering rates in typical applications, using the Inline-Plugins *CosmoPlayer 1.02* (VRML 1.0/2.0) and *DocShow-VR 0.9*. Here we show results from measurements with the 3D scene „DX-03“ that is distributed with the OpenGL benchmark *viewperf* [16]. The object was first converted from the viewperf-specific MSH (triangle-mesh) format to VRML 1.0/2.0 and DVR formats – DVR optionally with or without triangle-strip optimization. „DX-03“ represents a model that was

generated by IBM Data Explorer, a scientific visualization software, and consists of 91584 triangles as triangle-strips with ever ca. 100 triangles. The tests can be reproduced via this web page:

<http://www.dfn-expo.de/Technologie/DocShow-VR/dx.html>

It could be shown that *DocShow-VR* produces rendering rates similar to those of *viewperf*. But the triangle-strip optimization of the VRML-to-DVR converter had to be processed, and the plugin options *Normalize Normals*, *Two-sided Lighting*, and *Transparency* had to be disabled to reach this. The plugin options can be switched on or off by using popup menus, or by specifying them in the EMBED-tag that produces the 3D picture in the HTML page. Are these optimization facilities noticed, disadvantages regarding data volumes, startup times and rendering performance are established (see Table 1).

**Table 1** Performance comparison: DX-03 (512 x 512) on an SGI Onyx Reality Engine<sup>2</sup> (2 x R4400, 200 MHz, 2 RMs), accessing an Apache WWW server on an SGI Challenge L (2 x R4400, 200 MHz), connected via ATM (155 Mbps)

	DocShow-VR 0.9 / DVR Format		CosmoPlayer 1.02 / VRML 2.0	
	Using triangle strips	Without triangle strips	uncompressed	compressed (gzip)
Data volume [bytes]	2.252.744	6.601.928	9.768.093	2.266.695
Startup time (not progressive: without rendering)	0,795 s, progressive: 0,911 s	1,709 s, progressive: 1,894 s	ca. 25 s	ca. 25 s
Equivalent bitrate	22,7 Mbit/s, progr.: 19,8 Mbit/s	30,9 Mbit/s, progr.: 27,9 Mbit/s	3,1 Mbit/s	0,73 Mbit/s
Rendering time	0,155 s, optimized: 0,084 s	0,151 s, optimized: 0,144 s	ca. 0,36 s	
Rendering rate [triangles/s]	590.864, opt.: 1.090.286	606.517, optimized: 636.000	254.400	

## Results

The data volume of the optimized DVR file is similar to the gzip-compressed VRML file.

The startup times in this test case were reduced to 1/28.

The rendering rate differs in this test case in DocShow-VR around the factor 2 and is at best 4 times better than the CosmoPlayer. The peak polygon rate of the applied Reality Engine<sup>2</sup> graphics subsystem, as specified in the data sheet from SGI – „3D triangle meshes, smooth shaded, z-buffered, phong lighting: 1,07 Mio. triangles/s“ – is achieved. Nevertheless, these data and rendering optimizations are not generally applicable, since other 3D objects are partly not automatically optimizable, or in certain applications inefficient graphics attributes have to be used.

## 6 ACKNOWLEDGEMENTS

This work is partly funded by the DFN-Verein (German Research Network), with means of the BMBF (German Ministry for Education, Science, Research, and Technology) under the project „DFN-Expo“. The authors would like to thank C. Grimm and J.-S. Vöckler (RVS) for the discussion about and configuration of a converting proxy/cache-server prototype.

## 7 REFERENCES

1. Bell, G., Parisi, A., Pesce, M.: *The Virtual Reality Modeling Language – Version 1.0 Specification*. 09.11.1995.  
(<http://www.vrml.org/Specifications/VRML1.0/>)
2. Berners-Lee, T., Connolly, D.: *Hypertext Markup Language – HTML 2.0*. RFC 1866, 03.11.1995. (<ftp://nis.nsf.net/documents/rfc/>)
3. Berners-Lee, T., Masinter, L., McCahill, M.: *Uniform Resource Locators (URL)*. RFC 1737, 20.12.1994. (<ftp://nis.nsf.net/documents/rfc/>)
4. Brutzman, D., Zyda, M., Watsen, K., Macedonia, M.: *Virtual Reality Transfer Protocol (vrtp) Design Rationale*. Workshop on Enabling Technology: Infrastructure for Collaborative Enterprises (WET ICE): Sharing a Distributed Virtual Reality, MIT, 18.–20.06.1997.  
([http://www.stl.nps.navy.mil/~brutzman/vrtp/vrtp\\_design.ps](http://www.stl.nps.navy.mil/~brutzman/vrtp/vrtp_design.ps))
5. Fielding, R., Gettys, J., Mogul, J., Nielsen, H., Berners-Lee, T.: *Hypertext Transfer Protocol – HTTP/1.1*. RFC 2068, 03.01.1997. (<ftp://nis.nsf.net/documents/rfc/>)
6. Freed, N., Borenstein, N.: *Multipurpose Internet Mail Extensions (MIME)*. RFC 2049, 02.12.1996. (<ftp://nis.nsf.net/documents/rfc/>)
7. Hardenberg, J.: *RE: QvLib questions*. VRML Hypermail Archive, 27.03.1995.  
(<http://vag.vrml.org/www-vrml/arch/1107.html>)
8. Has, M., Newman, T.: *Color Management: Current Practice and The Adoption of a New Standard*, 1996. (<http://www.color.org/overview.html>)
9. ISO/IEC 14772-1: *The Virtual Reality Modeling Language (VRML97) – Part 1: Functional specification and UTF-8 encoding*. International Standard, 1997.  
(<http://www.vrml.org/Specifications/VRML97/>)



10. ISO/IEC 14772-3: *The Virtual Reality Modeling Language (VRML97) – Part 3: Compressed Binary Format Specification*. Editor's Draft 5, 1997.
11. Klein, R.: *Multiresolution representations for surface meshes*. In: Proceedings of the SCCG, 1997.  
(<http://www.gris.uni-tuebingen.de/people/staff/reinhard/mai97.ps.gz>)
12. Nebel, E., Masinter, L.: *Form-based File Upload in HTML*. RFC 1867, 07.11.1995. (<ftp://nis.nsf.net/documents/rfc/>)
13. Neider, J., Davis, T., Woo, M.: *OpenGL Programming Guide – The Official Guide to Learning OpenGL, Release 1*. Addison-Wesley, 1993.
14. Netscape: *Netscape Navigator LiveConnect/Plug-In Software Development Kit*, 1998. ([http://home.netscape.com/comprod/development\\_partners/plugin\\_api/index.html](http://home.netscape.com/comprod/development_partners/plugin_api/index.html))
15. Netscape: *Plug-In Guide – Communicator 4.0*. January 1998.  
(<http://developer.netscape.com/docs/manuals/communicator/plugin/>)
16. OPC – The OpenGL Performance Characterization Projekt: *Viewperf Information and Results*. (<http://www.specbench.org/gpc/opc.static/viewin~1.html>)
17. Paul, B.: *The Mesa 3-D graphics library*.  
(<http://www.ssec.wisc.edu/~brianp/Mesa.html>)
18. Schulzrinne, H., Rao, A., Lanphier, R.: *Real Time Streaming Protocol (RTSP)*. RFC 2326, 14.04.1998. (<ftp://nis.nsf.net/documents/rfc/>)
19. SDSC: *VRML Repository*. (<http://www.sdsc.edu/vrml/>)
20. Strauss, P., Bell, G.: *The VRML Programming Library – QvLib, Version 1.0 beta 1*. 1995. (<http://vag.vrml.org/www-vrml/vrml.tech/qv.html>)
21. Taubin, G., Rossignac, J.: *Geometric Compression Through Topology Surgery*. IBM Research technical report RC-20340, 16.01.1996.  
(<http://www.research.ibm.com/vrml/binary/pdfs/ibm20340.pdf>)
22. Upson, C., Faulhaber, T., Kamins, D., Laidlaw, D., Schlegel, D., Vroom, J., Gurwitz, R., van Dam, A.: *The Application Visualization System: A Computational Environment for Scientific Visualization*. In: IEEE Computer Graphics and Applications, July 1989.
23. Vöckler, J.-S.: *A quick glance at webfilt*. RVS, University Hannover, 03.09.1997. ([voeckler@rvs.uni-hannover.de](mailto:voeckler@rvs.uni-hannover.de))

# On the Optimal Placement of Web Proxies in the Internet: The Linear Topology

*Bo Li, Xin Deng and Mordecai J. Golin*

*Department of Computer Science, Hong Kong University of Science and Technology*

*Clear Water Bay, Kowloon, Hong Kong.*

*Tel: +852 2358 6976, Fax: +852 2358 1477*

*E-Mail: {bli, dengxin, golin}@cs.ust.hk*

*Kazem Sohraby*

*Bell Laboratories, Lucent Technologies*

*101 Crawfords Corner Road, Holmdel, NJ 07733, USA.*

*E-Mail: sohraby@lucent.com*

## Abstract

Web caching or web proxy has been considered as the prime vehicle to cope with the ever-increasing demand for information retrieval over the Internet, WWW being a typical example. The existing work on web proxy has primarily focused on content based caching; relatively less attention has been given to the development of proper placement strategies for the potential web proxies in the Internet. This paper investigates the optimal placement policy of web proxies for a target web server in the Internet. The objective is to minimize the overall latency of searching the target web server subject to the network resources and traffic pattern. Specifically, we are interested in finding the optimal placement of multiple web proxies ( $m$ ) among the potential sites ( $n$ ) under a given traffic pattern. We model the problem as a *Dynamic Programming* problem, and we obtain an optimal solution for a linear array topology using  $O(n^2m)$  time.

## Keywords

Web caching, Proxy server, Dynamic Programming

## 1 INTRODUCTION

We have witnessed an explosive growth in the use of World Wide Web (or web) in the past few years; there are many reasons behind this success, in particular, ease of use, the availability of standard tools for creating web documents and

for navigating the web, timely dissemination of information, and the increased popularity of the Internet [1]. At the same time, this quick adoption also leads to its poor performance, as web clients often have to tolerate long response times. There are a number of factors contributing to this inefficiency including server congestion during peak time, links with limited inadequate bandwidth, and long propagation delay. Caching has been considered as the prime vehicle to cope with this inefficiency.

The caching technique has been successfully used in the memory hierarchy [12] and distributed file system, AFS being one of such examples [10]. The basic principle behind caching is that it allows the retrieved documents to be kept close to the clients, this is essential in bringing down the access latency. There are several ways that documents can be cached for a web server including, web browser (client), web server itself and web proxy [15]. Caching at the clients' side has been implemented by most existing web browsers [2]. This can prevent a client from generating traffic to the same location repeatedly; for example both NCSA Mosaic and Netscape can save images and documents. Caching can also be deployed at the server side when a server contains pointers to other servers [15], this allows a web server to use a local copy fetching in advance to serve clients' requests, instead of having to forward the requests to remote server(s) each time. Unfortunately, both do little towards reducing the overall latency on the network [13]. Client side caching only saves one single client from fetching this document. In other words, each client has to cache the document, even if multiple clients accessing the same remote web document belong to the same local area network. Server caching only mitigates the problem of not forwarding requests further, but does nothing to alleviate the long access delay to the sever experienced by clients.

The most effective way in reducing the overall latency is the use of web proxy, or proxy server (or simply proxy). Web proxy is an intermediate server acting as an caching agent between clients and server. If properly designed, proxy can eliminate the possibly long propagation delay, and alleviate the potential inadequate link bandwidth path(s). Additionally, it also can reduce the server load, which may be critical during peak time. There has been considerable work on various aspect of web proxy, for example, traffic characterization [1], cache replacement algorithms [13] and server design [4, 9].

The effectiveness of proxy is primarily determined by *locality*, the same as for any cache. This locality depends a number of factors such as access patterns and configurations. The unique characteristics of web caching, *different* from conventional caching used in memory and distributed systems, is that locality is also largely influenced by the location of the web proxies. Simply put, placing a web proxy in the "wrong" place is not only costly, but also does little to improve the performance. In addition, it has also been shown that multiple web proxies are sometime needed in order to increase this locality, e.g., the hierarchical caching proposed in [4, 6].

Finding the optimal placement of web proxies in a network like the Internet

is a challenging task, as there is relatively little data on how well web proxy works. The decentralized and dynamic nature of the web adds extra complexity to this task [15]. Most existing proxies are placed in fairly “obvious” spots, e.g., gateway for a LAN, or some “strategic” locations [11]. To the best of our knowledge, there has been no systematical study on the proper placement of web proxies, which is the aim of this paper.

In this paper, we focus on two factors: the overall traffic and latency as described in [15]. The objective is to minimize the overall latency of searching the target web server subject to the network resources and traffic pattern. Specifically, we are interested in finding the optimal placement of multiple web proxies ( $m$ ) among the potential sites ( $n$ ) under a given traffic pattern. This turns out to be a very difficult problem, mainly caused by the dependency among the potential sites. This is because a potential site, say  $i$ , can be in place between another potential site ( $j$ ) and the web server. We define site  $i$  to be upstream of site  $j$  and  $j$  to be downstream of site  $i$ . The caching at any downstream site ( $j$ ) in general modifies the traffic pattern of the upstream site ( $i$ ). Unless the paths from all sites to the server are *disjoint*, in which the finding the optimal location becomes trivial, these dependencies significantly complicate the problem. In this paper, we consider a simple linear array topology. We show that this can be modeled as a *dynamic programming* problem, we further obtain the optimal solution for the linear array topology using  $O(n^2m)$  time.

The rest of the paper is organized as follows. We present the problem formulation in the next Section. Results are discussed in Section 3. We conclude the paper in Section 4 with discussions of on-going work.

## 2 PROBLEM FORMULATION

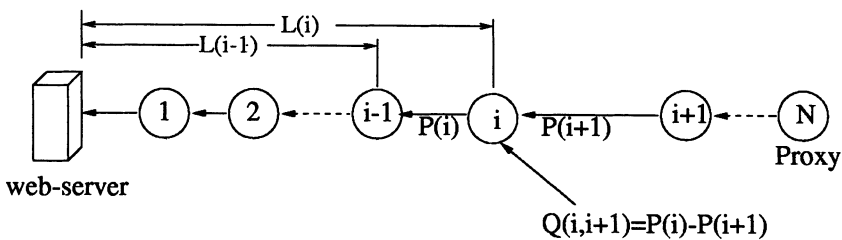


Figure 1 The Linear Configuration

We consider a one-dimensional array illustrated in Figure 1. Denote the potential web proxy locations by  $n = 1, 2, \dots, N$ . Without loss of generality, we assume the locations starting from the web server to be labeled as  $1, 2, \dots, N$ , i.e.,  $i$  and  $i+1$  are neighbors and  $i$  is closer to the web server than  $i+1$ . Let

$P(i)$  be the percentage of the overall traffic accessing the web that has to pass through node  $i$ . Since requests passing through node  $i$  must also pass through node  $i - 1$  we have  $P(1) \geq P(2) \geq \dots \geq P(N)$ . Let the propagation delay (distance) from node  $i$  to the server be  $L(i)$ ; If caching is done at the node  $i$ , we define the Gain to be  $G(i) = L(i) \times P(i)$  \*. This makes intuitive sense in that the percentage of the traffic ( $P(i)$ ) would not need to traverse the distance from the node  $i$  to the web server, i.e.,  $L(i)$ . We are interested in finding  $M$  ( $M \leq N$ ) web locations  $K_1, K_2, \dots, K_M$ , i.e.,  $K_1 < K_2 < \dots < K_M$ , that maximize the following value

$$(P(K_1) - P(K_2))L(K_1) + (P(K_2) - P(K_3))L(K_2) + \dots \\ + (P(K_{M-1}) - P(K_M))L(K_{M-1}) + P(K_M)L(K_M)$$

The main complication is dependency between the potential web proxies, specifically the caching at a node  $i$  will affect the up-stream node,  $1, 2, \dots, i-1$ . To simplify the notation, It will help us to slightly rewrite this problem. Set  $P(N+1) = 0$  and now, for  $1 \leq i \leq j \leq N+1$  define

$$Q(i, j) = P(i) - P(j) \quad i < j$$

Note that  $P_i = P_i - P_{N+1} = Q(i, N+1)$ .  $Q(i, j)$  is the amount of traffic coming to node  $i$  when node  $j$  has been chosen as one of the proxies and no nodes between  $i$  and  $j$  (i.e.,  $i+1, \dots, j-1$ ) are proxies;  $Q(i, i+1)$  is simply the traffic arriving at node  $i$  that did not pass through node  $j$  as illustrated in Figure 1b. Using this notation the expression that we wish to maximize becomes

$$L(K_1)Q(K_1, K_2) + L(K_2)Q(K_2, K_3) + \dots \\ + L(K_{M-1})Q(K_{M-1}, K_M) + L(K_M)Q(K_M, N+1)$$

To efficiently solve this optimization problem we will first generalize it

**Definition** Let  $n, m$  be such that  $2 \leq n \leq N+1$  and  $n-1 \leq m \leq M$ . Let  $K_1, K_2, \dots, K_m$  be such that  $1 \leq K_1 < K_2 < \dots < K_m < n$ . Set

$$\text{cost}(m, n : K_1, \dots, K_m) = L(K_1)Q(K_1, K_2) + L(K_2)Q(K_2, K_3) + \dots \\ + L(K_{m-1})Q(K_{m-1}, K_m) + L(K_m)Q(K_m, n)$$

The  $(m, n)$ -optimization problem is to find  $K_1 < K_2 < \dots < K_m$  that maximizes  $\text{cost}(m, n : K_1, K_2, \dots, K_m)$ .

Note: the reason for restricting  $n-1 \leq m \leq M$  is that in the  $(m, n)$  problem

---

\*The calculation derived in this paper also applies to other cost functions.

we must have  $1 \leq K_1 < K_2 < K_3 < \dots < K_m < n$ . If  $m < n - 1$  this is obviously impossible.

The original problem becomes the problem of maximizing  $\text{cost}(M, N + 1 : K_1, K_2, \dots, K_M)$ , i.e., solving the  $(M, N + 1)$  optimization problem. We will now develop a dynamic programming method that permits solving all of the  $(m, n)$ -optimization problems with  $2 \leq n \leq N + 1$  and  $n - 1 \leq m \leq M$ . Solution of the  $(M, N + 1)$  problem yields the solution to the original problem.

Our main observation is that solutions to the  $(m, n)$  problem must contain optimal solutions to certain subproblems.

**Lemma 1** *Let  $2 \leq n \leq N + 1$  and  $n - 1 < m \leq M$ . Further suppose that  $m > 1$ .  $K_1, K_2, \dots, K_m$  is an optimal solution to the  $(m, n)$  problem then  $K_1, K_2, \dots, K_{m-1}$  is an optimal solution to the  $(m - 1, K_m)$  problem.*

**Proof:** Suppose, by contradiction that the lemma is incorrect. Then there exist  $m, n, K_1, K_2, \dots, K_m$  and  $K'_1, K'_2, \dots, K'_{m-1}$  such that  $K_1, K_2, \dots, K_m$  solves the  $(m, n)$  optimization problem but  $K_1, K_2, \dots, K_{m-1}$  does not solve the  $(m - 1, K_m)$  one because

$$\text{cost}(m - 1, K_m : K'_1, K'_2, \dots, K'_{m-1}) < \text{cost}(m - 1, K_m : K_1, K_2, \dots, K_{m-1}).$$

But then

$$\begin{aligned} & \text{cost}(m, n : K'_1, \dots, K'_{m-1}, K_m) \\ &= L(K'_1)Q(K'_1, K'_2) + L(K'_2)Q(K'_2, K'_3) + \dots \\ & \quad + L(K'_{m-1})Q(K'_{m-1}, K_m) + L(K_m)Q(K_m, n) \\ &= \text{cost}(m - 1, K_m : K'_1, \dots, K'_{m-1}) + L(K_m)Q(K_m, n) \\ &< \text{cost}(m - 1, K_m : K_1, \dots, K_{m-1}) + L(K_m)Q(K_m, n) \\ &= L(K_1)Q(K_1, K_2) + L(K_2)Q(K_2, K_3) + \dots \\ & \quad + L(K_{m-1})Q(K_{m-1}, K_m) + L(K_m)Q(K_m, n) \\ &= \text{cost}(m, n : K_1, \dots, K_{m-1}, K_m) \end{aligned}$$

contradicting the optimality of  $K_1, K_2, \dots, K_m$  for the  $(m, n)$  problem.

This leads immediately to the following corollary:

**Corollary 2** *Let  $K_1, K_2, \dots, K_m$  be an optimal solution to the  $(m, n)$  problem and  $K'_1, K'_2, \dots, K'_{m-1}$  be an optimal solution to the  $(m - 1, K_m)$  problem. Then  $K'_1, K'_2, \dots, K'_{m-1}, K_m$  is also an optimal solution to the  $(m, n)$  problem.*

**Proof:** From the Lemma we already know that  $K_1, K_2, \dots, K_{m-1}$  is also an optimal solution to the  $(m - 1, K_m)$  problem and thus

$$\text{cost}(m - 1, K_m : K_1, \dots, K_{m-1}) = \text{cost}(m - 1, K_m : K'_1, \dots, K'_{m-1}).$$

Therefore

$$\begin{aligned}
 & \text{cost}(m, n : K_1, \dots, K_m) \\
 = & \text{cost}(m, K_m : K_1, \dots, K_{m-1}) + L(K_m)Q(K_m, n) \\
 = & \text{cost}(m, K_m : K'_1, \dots, K'_{m-1}) + L(K_m)Q(K_m, n) \\
 = & \text{cost}(m, n : K'_1, \dots, K'_{m-1}, K_m)
 \end{aligned}$$

Since  $K_1, \dots, K_{m-1}, K_m$  is optimal for  $(m, n)$  and  $K'_1, \dots, K'_{m-1}, K_m$  has the same cost as

$K_1, \dots, K_{m-1}, K_m$  this implies that  $K'_1, \dots, K'_{m-1}, K_m$  is optimal as well.

Now define the following

**Definition** Let  $2 \leq n \leq N + 1$  and  $n - 1 \leq m \leq M$ . Then

$OPT(m, n) =$  maximal value solution for the  $(m, n)$  problem.

If  $m = 1$  then, by definition,

$$OPT(1, n) = \max_{1 \leq k < n} L(k)Q(k, n). \quad (1)$$

If  $m > 1$  and  $K_1, K_2, \dots, K_m$  is a solution to the  $(m, n)$  problem then, from Lemma 1,  $K_1, K_2, \dots, K_{m-1}$  is a solution to the  $(m - 1, K_m)$  subproblem so

$$\begin{aligned}
 OPT(m, n) &= \text{cost}(m, n : K_1, \dots, K_{m-1}, K_m) \\
 &= \text{cost}(m - 1, K_m : K_1, \dots, K_{m-1}) + L(K_m)Q(K_m, n) \\
 &= OPT(m - 1, K_m) + L(K_m)Q(K_m, n)
 \end{aligned}$$

Thus, for  $m > 1$ ,

$$OPT(m, n) = \max_{1 \leq k < n} [OPT(m - 1, k) + L(k)Q(k, n)]. \quad (2)$$

Equations (1) and (2) can together be used to calculate all of the  $OPT(m, n)$  values. To also calculate the actual solution locations we define another array  $K(m, n)$  that satisfies  $K(1, n) = k$  such that

$$k < n \text{ and } L(K)Q(k, n) = OPT(1, n)$$

i.e.,  $K(1, n)$  is a solution location for the one-cache problem and  $K(m, n) = k$  such that

$$k < n \text{ and } OPT(m - 1, K_m) + L(K_m)Q(K_m, n) = OPT(m, n)$$

i.e.,  $K(m, n)$  is a rightmost cache location in some solution to the  $(m, n)$

problem. Notice that if there are many solutions,  $K(m, n)$  might not necessarily be uniquely defined. Given this  $K(m, n)$  array we can calculate a solution by setting  $K_m = K(M, N + 1)$  and, iteratively, for all  $i < m$ , setting  $K_i = K(i, K_{i+1})$ . Repeated applications of Corollary 2 shows that

$$\text{cost}(m, n : K_1, \dots, K_m) = \text{OPT}(M, N + 1)$$

and is thus the solution we are looking for.

Pseudocode for constructing the  $\text{OPT}()$  and  $K()$  arrays is given below. The first, initialization, section, has two  $O(n)$  size nested for loops and therefore uses  $O(n^2)$  time in total. The second section contains an outer  $O(m)$  size for loop each iteration of which calls two nested  $O(n)$  size loops. Thus the entire section, and therefore the entire algorithm, runs in  $O(n^2m)$  time.

### 1. Initialization.

```

for  $n := 2$  to  $N + 1$  do
     $k := 1$ ;
    for  $j := 2$  to  $n - 1$  do
        if  $L(j)Q(j, n) > L(k)Q(k, n)$  then  $k := j$ 
     $\text{OPT}(1, n) := L(k)Q(k, n)$ ;  $K(1, n) := k$ ;
```

### 2. Filling in the array

```

for  $m := 2$  to  $M$  do
    for  $n := m + 1$  to  $N + 1$  do
         $k = m$ ;
        for  $j := m + 1$  to  $n - 1$  do
            if  $(\text{OPT}(m - 1, j) + L(j)Q(j, n)) > (\text{OPT}(m - 1, k) + L(k)Q(k, n))$ 
                then  $k := j$ 
         $\text{OPT}(m, n) := \text{OPT}(m - 1, k) + L(k)Q(k, n)$ ;  $K(m, n) := k$ ;
```

## 3 RESULTS

In this section, we present an example to illustrate how the algorithm works. The example considers a configuration with  $N = 10$  and  $M = 5$ , shown in Figure 3. The delay time  $L(i)$  and traffic probabilities  $P(i)$  are given in Table 1. This example shows a case with balanced spread-out traffic, specifically, each of the 10 nodes contributes 10% traffic (i.e.,  $Q(i, i + 1)$ ). The distance is what dictates the measurement  $P(i)L(i)$ .

The algorithm essentially needs to fill in the matrix  $K(M, N + 1)$ , whose entry  $K(m, n)$  ( $m = 1, 2, \dots, M$ ,  $n = 2, 3, \dots, N + 1$ ) denotes the rightmost (farthest) proxy location, i.e.,  $K_m = K(m, n)$ , as defined earlier. Recall from Lemma 1, that the next element  $K_{m-1}$  must be the rightmost optimal solution



	1	2	3	4	5	6	7	8	9	10
$L(i)$	0.1	0.6	1.6	3.1	5.1	7.6	10.6	14.1	18.1	22.6
$P(i)$	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1

**Table 1** Example: The delay time  $L(i)$  and traffic percentage  $P(i)$

	1	2	3	4	5
10	(7)	(6, 9)	(5, 7, 9)	(4, 6, 8, 10)	(4, 6, 8, 9, 10)

**Table 2** Example: The optimal solution  $(M, N)$

for  $K(m-1, K_m)$ , i.e., the next element is  $K_{m-1} = K(m-1, K_m)$ . Accordingly, we can obtain all the optimal proxy locations :  $K_1 = K(1, K_2)$ ,  $K_2 = K(2, K_3)$ ,  $\dots$ ,  $K_{M-1} = K(M-1, K_M)$ ,  $K_M = K(M, N+1)$ .

The algorithm is divided into two parts: the first part is *initialization*, which calculates the  $K(1, N+1)$ . For example, in the above example,  $K(1, 11) = 7$ , simply because  $L(7)P(7) = 4.24$  is the maximum over all  $L(i)P(i)$   $i = 1, 2, \dots, 10$ .

The next step is to fill in the rest of the elements in  $K(m, n)$ . For the above example, referring to the Figure 2. That  $K(5, 11) = 10$  means that  $K_5 = 10$  is the rightmost optimal proxy location. The next one is therefore  $K(5-1, 10) = K(4, 10) = 9$ , and so on. We have the optimal solution is (4, 6, 8, 9, 10) shown in Table 2 and Figure 3.

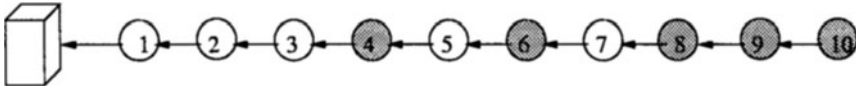
Notice, as defined in Section 2, that the notation  $K(m, n)$  in Figure 2 assumes the last element ( $n$ ) has no traffic. For the rightmost proxy location ( $K_M$ ), this is obtained by calculating  $K(M, N+1)$ . For others  $K_i = K(i, K_{i+1})$ , since the  $K_{i+1}$  has been chosen as one of the proxy locations, Hence the traffic generated from location  $K_{i+1}$  to location  $K_i$  is also zero. This is precisely what the term  $Q(i, j)$  does.

## 4 CONCLUSION

In this paper, we investigate the optimal placement policy of web proxies for a target web server in the Internet. The objective is to minimize the overall latency of searching the target web server subject to the network resources and traffic pattern. Our contributions are 1) formulating the problem into a *dynamic programming* problem; 2) obtaining an optimal solution for a *linear array topology* using polynomial time.

$\begin{smallmatrix} M \\ \diagdown \\ N \end{smallmatrix}$	1	2	3	4	5
1					
2	1				
3	2	2			
4	3	3	3		
5	3	4	4	4	
6	4	5	5	5	5
7	5	6	6	6	6
8	5	6	7	7	7
9	6	7	8	8	8
10	7	8	8	9	9
11	7	9	9	10	10

**Figure 2** Example: The optimal solution matrix  $I K(M, N + 1)$



**Figure 3** Example: The optimal proxy locations

The model proposed here can be easily extended to handle the following two cases:

- Hierarchical caching, in which the down-stream proxies only hold a subnet of the documents of the up-stream proxies. In such cases each down-stream proxy can only block a portion of the traffic. This can be handled by re-defining the notation  $P(i)$  to be only the percentage of the traffic that the  $i^{th}$  node's cache can serve.
- Different link bandwidth. This can be dealt with by incorporating the link bandwidth into the distance  $L(i)$ , e.g., assigning larger value to slower links.

We are currently working on refining the model by relaxing two key assumptions in this paper, *linear topology* and *static traffic pattern*. In particular, we are studying a *tree topology*, which is considered to be more realistic topology for the Internet. The web server is the root of the tree. The complication, similar to the linear topology, is dependency among the potential web proxies. The result we obtained recently demonstrates that this can also be modeled as a dynamic programming problem with higher complexity  $O(n^3 m^2)$  [8]. In

addition, we are also working on reducing this complexity, and the preliminary result indicates that this can be brought down to  $O(n^2m^2)$  [5].

The second issue concerns the dynamic nature of the traffic. The placement policy ideally should be distributed and adaptive. We are investigating this issue in the content of *active networking* [14]. The model assumes the potential web proxy sites can somehow monitor the traffic periodically, which is one of properties within an active network environment, and then make the decision about whether caching or not based on a threshold. Specifically, if the observed traffic volume within an observed period is above the threshold, the potential site will cache the web content. Notice that the threshold is determined by a number of factors, in particular the distance  $L(i)$ . In other words, different sites have different thresholds. This makes intuitive sense in that the closer the potential site is to the target web server, the higher the threshold should be. The results show that such distributed decisions can potentially lead to convergence to the static solution proposed in this paper by properly selecting the threshold [7].

The future work will consider more realistic web traffic distribution, for example capturing the actual workload characterization [1], or considering the Zipf distribution used by Bestarov [3].

## REFERENCES

- [1] M. F. Arlitt and C. L. Williamson, "Internet Web Servers: Wordload Characterization and Performance Implications," *IEEE Transactions on Networking*, Vol. 5, No. 5, October 1997.
- [2] M. Baentsch, L. Baum, G. Molters. S. Rothkugel and P. Sturm, "World Wide Web Caching: The Application-Level View of the Internet," *IEEE Communications Magazine*, Vol. 35, No. 6, June 1997.
- [3] A. Bestavros, "WWW Traffic Reduction and Load Balancing Through Server-based Caching," *IEEE Concurrency*, Vol. No. , January 1997.
- [4] S. Glassman, "A Caching Relay for World Wide Web," *Computer Networks and ISDN Systems*, Vol. 27, No. 2, November 1994.
- [5] M. Golin, G. Italiano and A. Vigneron, "The p-median problem on directed trees," To be submitted for publication.
- [6] C. Bowman, P. Danzig, D. Hardy, U. Manber and M. Schwartz, "The Harvest Information Discovery and Access System," *Computer Networks and ISDN Systems*, Vol. 28, No. 1-2, December 1997.
- [7] B. Li, X. Deng, M. J. Golin and K. Sohraby, "Dynamic and Distributed Web Caching in Active Networks," Submitted to *APWeb'98*, April 1998. Also Li's presentation at Bell Lab, Lucent Technology, June 1997.
- [8] B. Li, M. J. Golin, G. Italiano, X. Deng and K. Sohraby, "On The Optimal Placement of Web Proxies in the Internet," Submit to *IEEE Transactions on Knowledge and Data Engineering: Special Issue on Web Technologies*, May 1998.

- [9] A. Luotonen and K. Altis, "World Wide Web Proxies," *Computer Networks and ISDN Systems*, Vol. 27, No. 2, November 1994.
- [10] J. Morris, "Andrew: A Distributed Personal Computing Environment," *Communications of ACM*, Vol. 29, No. 3, March 1986.
- [11] M. Nabeshima, "The Japan Cache Project: An Experiment on Domain Cache," *Computer Networks and ISDN Systems*, Vol. 29, No. 8-13, September 1997.
- [12] D. Patterson and J. Hennessy, *Computer Organization and Design: the Hardware/Software Interface*, 2nd edition, *Morgan Kaufman*, 1997.
- [13] P. Scheuermann, J. Shim and R. Vingralek, "A Case for Delay-Conscious Caching of Web Documents", *Computer Networks and ISDN Systems*, Vol. 29, No. 8-13, September 1997.
- [14] D. Tennenhouse, J. Smith, W. Sinncoskie, D. Wetheral and G. Minden, "A Survey of Active Network Research," *IEEE Communications Magazine*, Vol. 35, , No. 1, January 1997.
- [15] N. Yeager and R. McGrath, *Web Server Technology*, *Morgan Kaufman*, 1996.

# The Network Computer for an Open Services Market

*Lutz Henckel, Jiri Kuthan*

*GMD FOKUS*

*Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany,*

*{lutz.henckel,kuthan}@fokus.gmd.de*

## **Abstract**

The approach of a Network Computer (NC) benefits from a client/server architecture using distributed computing capabilities but avoids the disadvantages of the required maintenance for distributed operating system and application software. In general today's available NC solutions are only applicable in an Intranet environment. The maintenance of NC operating system and applications is carried out on a central server and the NCs download the current software from the server if needed. The required management of users, storage, internet access and application usage is performed by a human administrator, who has to set up corresponding configurations manually.

Through effort centralisation for operation system and application maintenance the NC users are relieved from this task. Additionally an interconnected client/server architecture opens the possibility to access and use resources (e.g. storage, e-mail, www, ftp, internet access, etc.) over the network. These are the main reasons why enhancements of this NC approach are proposed. In doing so any provider may offer its services on an open market. The offered services can be subscribed and used by NC users acting as customers. The required enhancements encompass the definition and realisation of an Internet Services Management Protocol (ISMP) supporting dynamic negotiation of the customer/provider relationship without administrator interactions.

Besides the usage of NCs in a Small Office/Home Office (SOHO) area, it is imaginable to install corresponding equipment at public locations like airports, railroad stations, restaurants, hotels, telephone boxes, etc. Especially the mobility

of NC users is supported by the enhanced approach because the users get their well-known and customized environment independent from the location where an NC is used.

### **Keywords**

Network Computer, Java, Open Services Market, Internet

## **1 JAVA: WRITE ONCE/RUN ANYWHERE**

The basics for a realization of Network Computers is Java [1]. Following the principal Write once/ Run anywhere the Java Development Environment (JDK) supports the realization of programs which can be executed on almost all computer or operating system platforms. Assumption for that is the availability of a Java Virtual Machine (JVM) on a target system, which is able to interpret the Byte Code generated by a Java Compiler. The JVM transforms the Byte Code into machine dependent code and then executes the program. Assuming that software has to be developed only for this virtual Java-Platform to make it executable on any system the porting effort for other platforms can be avoided. On one hand this reduces the software development costs substantially and opens smaller software manufacturers better sales opportunities. On the other hand users gain freedom during choice of a computer and operating system platform.

The second important assumption for a realization of the NC approach is the possibility to download Java-based applications over the internet for local execution. This is supported by the features to embed Java-Applets in HTML documents, which will be loaded if a user accesses the corresponding web page. Furthermore today's Web-Browsers contain a JVM as an integral component providing the direct execution of currently downloaded Java-applications. Especially these facilities open new perspectives for software distribution and maintenance.

## **2 NC FOR THE INTRANET**

The client/server architecture is the most disseminated architecture for company internal computer solutions as well as the current expansion rate for the usage of internet protocols in this area indicates, that the Intranet architecture is the most promising one for the future. Today's company internal networks often consist of PCs running MS-Windows and Unix-based Servers. Besides undeniable advantages against a centralized architecture also cost disadvantages have been exposed. About 12.000 \$ per year have to be calculated for the administration and maintenance of the PC operating system and applications for these Fat Clients. This was a result of a study about Management Strategies to Control the Rapidly Escalating Costs of Distributed Computing [3] done by the Gartner Group.

Lowering of costs towards 2.500 \$ per workstation can be achieved by using Thin Clients [2]. The advantage of a client/server architecture with the availability of computing capacity at any workstation remains. On the other hand the applications are administrated at a central server and are downloaded only if needed. Using Java-Technology applications may be executed on almost all platforms. As a result the effort for distribution and maintenance of applications and the operating system for every workstation is no longer necessary. Nevertheless the savings of maintenance efforts and corresponding costs have to be compared with additional needs on communication resources and costs.

In may 1996 Sun, Oracle, Netscape, IBM and Apple (SONIA) have jointly presented the Network Computer (NC) Reference Profile [4], which determines a set of Internet Standards an NC has to support. At the beginning of 1997 first NC products were introduced by IBM and Sun which can only be used in an Intranet environment. The NC equipment only supports adapters for local area networks (e.g. Ethernet, Token Ring) and does not include a hard disk or other storage media. As a consequence of this architecture a download of the operating system is required each time the NC is switched on.

The creation of user accounts or mailboxes as well as the assignment of OS/File servers to NCs is performed furthermore by an administrator, who has to setup corresponding configurations manually on server side. After configuration completion the NC users may use the following services:

- Web Browsing
- File Access and Management
- File Transfer
- Send and Receive E-Mails
- Join Usenet Discussion Groups
- Naming/Directory Services
- Internet Access

### 3 NC-REQUIREMENTS FOR THE INTERNET

For the usage of NCs in an open services market environment some concepts have to be adapted. Especially the inflexible and slow management performed by a human administrator plays an important rule. Better flexibility and automation of these management functions and other following requirements have to be taken into account for a design of a system architecture for Network Computers in an open services market:

architecture which supports load balancing to avoid performance bottlenecks.

#### **Openness**

**On-line Service Subscription**

The registration of NC users as well as subscription and usage of services should be effected on-line without interaction of an administrator at service provider side. Customers buying an NC don't have to decide which service provider they are going to use. The NC user has the freedom to select one of different providers offering one or multiple services and may change the provider whenever desirable.

**Zero Maintenance**

The NC user should not need to pay attention to the maintenance of NC operating system (NCOS) or applications. This effort should be performed automatically by the service provider.

**Mobility: Subscribe once/Use everywhere**

Users may use NCs at home but also in remote regions at workplace or public locations. The customized working environment (file system, applications, mailboxes, etc.) of a user should be remained without any NC configuration changes.

**Network Access**

A SOHO-NC as well as a public NC needs a network adapter to access public networks like ISDN, GSM, ADSL, etc.

**Performance**

Assuming the usage of Public Networks only less bandwidth is available. This has to be taken into account and may not become a knock-out criterion for the NC system architecture.

**Client/Server Independency**

Due to changing requirements the service provider needs the possibility to adapt its system configuration. This includes the support of a growing number of users, the provision of additional service access points in different regions, the relocation of user resources on other servers as a result of crashes or resource lacks. The adaptation of the provider-side configuration may not require configuration changes at NC side, which have to be performed by the NC user.

**Scalability**

The scalability of the server-side architecture has to be taken into account, so that a very large number (millions) of NC users can be supported by a service provider. The main problems in this area are the allocation of unique numerical user identifiers, the management of large databases for registered users and a distributed architecture which supports load balancing to avoid performance bottlenecks.

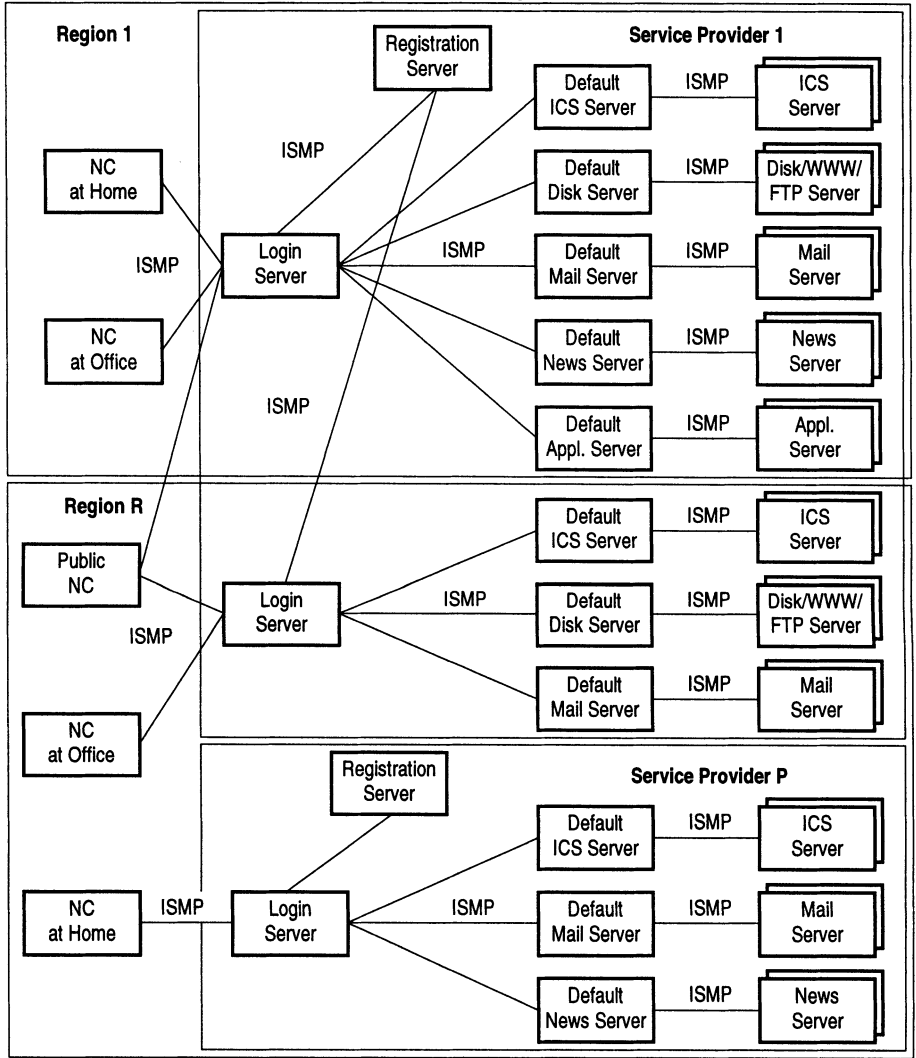
**Openness**



Prerequisite for the development of an open market for manufacturers of NC equipment as well as NC customers and service providers is the disclosure of used protocols and interfaces for services management and usage.

4 NC SYSTEM ARCHITECTURE FOR THE INTERNET

Considering the requirements defined above an enhanced client/server system architecture is introduced, which is shown in Fig. 1.



**Figure 1** NC System Architecture for an Open Service Market  
The enhanced services [5] provided in an open services market are the following:

- The **Internet Connectivity Service (ICS)** controls Internet access and provides a persistent user environment as well as an automatic NCOS update function.
- The **Disk Service** provides transparent access and management of remote disk space for the storage of private data. An optional backup function is included which is performed automatically by the Disk service provider.
- The **WWW Service** provides transparent access and management of remote disk space on a public accessible WWW server. An optional backup function is included which is performed automatically by the WWW service provider.
- The **FTP Service** provides transparent access and management of remote disk space on a public accessible FTP server. An optional backup function is included which is performed automatically by the FTP service provider.
- The **Mail Service** enables a user to send and receive e-mails.
- The **News Service** enables a user to retrieve and post articles of Usenet discussion groups.
- The **Application Service** enables a user to download and use applications.

Before an NC user is able to use a service a registration at a provider has to be done. Therefore the user has to specify its full name, address, bank detail as well as a user name and administration password which is needed to manage the services. Furthermore for each of the above introduced services five functions are supported:

- **Subscribe/Unsubscribe a service:** The NC user has to specify a service password which is needed for authentication during service login and the provider allocates the requested resources.
- **Login/Logout a service:** The NC user authenticates itself against the provider using its service password and the provider releases the resources for access.
- **Change service parameters:** The NC user may change service-specific parameters, e.g. service password, amount of allocated disk space, backup interval, etc.

The client-side deploys NCs enabling users to subscribe, login and use services offered by the providers at server-side. It is assumed that several competing service providers exist and each NC user may subscribe several services also from different providers. Larger providers will offer their services national or international wide by providing an adequate server infrastructure in each region. Therefore load balancing can be achieved and shorter transfer routes lead to reduced response times. Finally communication costs are reduced as a result of local reachable access nodes.

The actual usage of subscribed services is performed by the Internet protocols (SSAP - Service-Specific Access Protocols) which are defined in the NC Reference Profile [4]. Additionally the dynamic management negotiation between service user and provider is supported by the Internet Services Management Protocol (ISMP).

The enhanced NC system architecture consists of the following components:

The Network Computer (NC) is an economically priced system with CPU, memory, graphic, sound and network adapter as well as a keyboard, pointing device and monitor. Additionally a hard disk is included, which is used as a cache device for NCOS, HTML pages and images. The NC provides an easy-to-use graphical user interface allowing the user to register itself at different service providers as well as supporting service subscription and login. The NC Session Manager (SM) is the main application for the NC user to manage NC services. The management functions are realized by the SM using the Internet Services Management Protocol (ISMP, see below). The actual service usage is provided by other applications like a file and directory manager, web browser, ftp and e-mail clients as well as a news reader. But most of today's browsers integrate all these functions in one application, e.g. Netscape Communicator.

Using a local hard disk as a cache device a download of NCOS is not needed each time the NC is switched on. A download is only necessary if a new NCOS version is available from the service provider. As a result of this caching function the latency is reduced to a minimum which is especially important if narrowband networks are used (e.g. ISDN, GSM).

The Login Server (LS) controls the access to services of particular providers in a region. It verifies the access rights of NC users by comparing the specified user name and password with the entries of the central Provider Passwd Database which contains information about all registered users. To perform service management functions the SM of an NC communicates only with the LS, which forwards the orders to the servers (DSS - Default Service Server; SS - Service Servers) responsible for a particular service. The service provider has freedom to reconfigure its server-side infrastructure without performing any configuration effort at NC side. This is achieved by sending the address of the actual relevant service server within the reply each time a service login is initiated. This server address is then automatically used during service access initiated by SM at NC side.

The Registration Server (RS) is the central host of the service provider infrastructure where all databases reside containing information about registered NC users. The Provider Passwd Database encloses common data of NC users, e.g. full name, address, bank detail as well as user name and administration password. Additionally there exists a Service Passwd Database with service-specific data for every offered service or application containing the address of actual relevant service server and the service password. The other servers have the possibility to access these central databases (currently using NFS) to check service access rights of users or to manage user entries. The RS assigns a numerical user identification (UID) to every user, which is unique in the area the provider is responsible for. This is necessary to perform access control to file systems which are accessed and managed using NFS. An NCOS kernel-based mapping from local UID to remote ones and vice versa is necessary because a single user may get a different UID from each provider.

The Default Service Server (DSS) accepts the subscription orders for a given service from LS and selects a Service Server (SS) having enough resources to provide the requested service. Also compliance with performance requirements is taken into account during the selection process.

The Service Server (SS) establishes the service it is responsible for and makes it available for an NC user. The SS insures that the service can only be used if it was successfully subscribed and a user specifies correct user name and password during service login phase. During service subscription an user entry is included in the Service Passwd Database. Besides service-specific parameters each entry contains the host address of the actual relevant Service Server which is delivered to NC within the reply message of ISMP during service login. This allows the provider to relocate the resources of an NC user on another Service Server. Such a relocation may be necessary if a Service Server crashes or has to be maintained. The actual service access is performed using the Service-Specific Access Protocols (SSAP) defined in [4].

After an NC user performs a registration and authentication (login) at the Internet Connectivity Service (ICS) Server the access to local servers at provider side and to other servers of the global Internet is granted.

Additionally the NC user gets access via NFS to a remote directory on ICS Server to store personal configuration data. These data contain information about services subscribed at different providers including provider name, LS host address, user name and service password as well as additional service-specific information. The management of these data is performed by the Session Manager (SM) of NC. The data deposition on a server at provider side is required to keep the actual status of subscribed services (persistent) for the user on one hand and on the other hand to have the data available independently from the location the user is working with an NC.

Each time an NC user performs a login for the Internet Connectivity Service (ICS) the ICS server sends the version number of current NCOS. The SM compares this number with the number of the current running NCOS. If the running version is elder than the version on server side the user is prompted to give its confirmation for update. On acceptance the remote file system containing the new NCOS version is mounted by the NC, automatically installed on the NCs hard disk and finally the NC is rebooted with the new NCOS. Providing this centralized maintenance of NCOS the user is relieved from this task.

The Disk/WWW/FTP Service Server creates a directory on the servers file system for the NC user during subscription phase. After verification of users access rights during login phase the directory is exported to NC which mounts it on local file system. Therefore the NC user is able to access the remote directory via Network File System (NFS) protocol just like a file system mounted from a local hard disk.

The three server types differ in their access possibilities of other Internet users. The storage on a Disk servers hard disk is only accessible via NFS for the NC user who has subscribed the Disk Service. Therefore it is only applicable for storing

private data. In contrast to that the disk space of a WWW Service Server is simultaneously a part of the directory tree of a public accessible Web-Server. Accordingly a file system directory of an FTP Service Server is simultaneously a part of a public accessible FTP Server.

The amount of disk space which is requested by an NC user for allocation is signalled by ISMP. The Service Server maps the disk space parameters to corresponding values of the local quota system on server operating system. Therefore the quota system is used to check disk saturation and avoids violation of negotiated limits.

The Disk/WWW/FTP Service Server supports an additional backup function. During service subscription an NC user may specify a time interval which is signalled via ISMP. The service server automatically makes a backup of the directory depending on the requested interval.

The Mail Server establishes a mailbox for a user and configures an alias name, which may be specified by the user. The read access on a mailbox is supported by POP3/IMAP4 protocols and access rights are verified using the user name and password negotiated between SM and Mail Service Server during service subscription phase. Finally sending mail is supported by SMTP. The user rights for sending mail are checked using rules which are dynamically configured during login phase. For setting up these rules the negotiated user name and password are used as well as the NC host address.

The News Server controls the access to discussion groups including articles distributed by the Usenet system. The access rights of a registered user to retrieve and post articles are set if it has subscribed the News Service and performed a successful login. In the Usenet system corresponding rules have to be configured dynamically by the News Server to control these user access rights.

The Application Server controls the usage of applications which are subscribed by registered NC users. The java-based application resides in a directory of a WWW Server and may be downloaded by a browser using HTTP. The access to an application directory may only be granted for registered users which have performed a successful login. To insure this access control mechanisms of the WWW Server are used. For each application directory a database is configured with an entry for each user currently allowed to access the application. The user entries contain the user name and password negotiated via ISMP. An entry is created dynamically during login phase and is removed during logout. Each time a user accesses the application directory the WWW server asks for the user name and password. The access is granted only if a corresponding entry in the access control database was found.

Each time a user performs a login for a subscribed application the java applet is downloaded again. As a result the user can be sure that it is using the newest version available at this time. Providing this centralized maintenance of applications the user is relieved from this task.

The Internet Service Management Protocol (ISMP) is a TCP/IP based, transaction-oriented protocol for dynamic management of Internet services. It

allows users to subscribe services and controls their usage. Especially ISMP provides reliability and authentication features to avoid incorrect service charging which may occur as a result of transaction processing over the unreliable Internet. The ISMP is used for negotiation between customer and provider (namely between SM and LS) as well as for internal communication between provider's servers (LS, RS, DSS and SS). The LS and DSS act as a dispatcher forwarding protocol data units to other server entities.

The Service-Specific Access Protocols (SSAP) are standardized Internet protocols which are determined in the NC Reference Profiles [4]. The following protocols are used to access the subscribed services after a user has performed a successful login:

- HTTP for browsing the Web and usage of java-based applications
- NFS to manage remote file systems
- SMTP to send e-mail
- POP3/IMAP4 to receive e-mail
- NNTP to join Usenet discussion groups

## 5 NC PROTOTYPE REALIZATION

A prototype implementation based on the introduced NC architecture has been developed at FOKUS. Features as on-line service subscription, Web-driven server administration, automatic NC administration and reliable service charging are supported. Extensive tests with the prototype have proved the concept to be a reliable means for electronic service renting and management.

Solaris-based Sun workstations powered by Sparc processors have been employed as server systems. We have taken advantage of scalability features supported by the Solaris Operating System: wide uid range, dynamic setting-up kernel properties, and robustness of the operating system guarantees operation even under high load of many on-line users. The NC server components were written in ANSI C and the web-driven management software was implemented by the server-side HTML-embedded scripting language PHP/FI [6].

PC with Linux has been selected for the client side because it is an inexpensive platform for which all components required in NC Reference Profile [4] (internetworking protocols, Java support, hardware requirements, etc.) are available. The Session Manager (SM) providing an easy-to-use user interface for service management, has been written in Java to achieve maximum portability (only OS-specific procedures have been implemented using shell scripts). The well-known Netscape Communicator is used to access most services managed over ISMP (application-service, e-mail, news, www, ftp). Finally a file and directory manager may be used to manage the remote file systems exported from the servers providing disk, www and ftp services.

## 6 CONCLUSION

Current available Network Computer (NC) solutions are only applicable in an Intranet environment because the administration of users, storage, internet access and application usage is performed by a human administrator who has to set up corresponding configurations manually. This is inflexible and slow to be appropriate for an Open Services Market environment where customers may subscribe, use and change services offered by many competing service providers. Therefore the current NC approach was enhanced by a mechanism to negotiate the management parameters dynamically between customer NCs and the infrastructure of providers. This is necessary in order to automate the administration tasks for managing the relationship in an open client/server environment. The dynamic administration mechanism is realized by an Internet Services Management (ISMP) protocol, an NC Session Manager (SM) providing an easy-to-use user interface and administration servers at provider side.

Furthermore the proposed enhancements support provider-driven maintenance of the NC operating system, applications and other server-based resources. Without these maintenance tasks the usage of computers and the internet is much easier. This is the main reason why we believe to gain new customers with such an enhanced architecture, especially those people where the maintenance of Fat Clients expects too much of them or who are simply tired of wasting time with administration tasks. Additionally the concept of an open services market opens smaller companies the opportunity for outsourcing their server-based infrastructure. This avoids the needs of hiring expensive computer experts. Finally the availability of public NCs at airports, railroad stations, restaurants, hotels, telephone boxes, etc. will offer better personal mobility approximating a realization of the principal Subscribe once / Use everywhere.

## 7 REFERENCES

- [1] Gosling, J., McGilton, H., The Java Language Environment, White Paper, Sun Microsystems, 1996, [http://java.sun.com/nav/read/white\\_papers.html](http://java.sun.com/nav/read/white_papers.html).
- [2] Tribble, B., Java Computing in the Enterprise, What it means for the General Manager and CIO, White Paper, Sun Microsystems, 1996, <http://www.sun.com/javacomputing>.
- [3] Gartner Group, Management Strategies to Control the Rapidly Escalating Costs of Distributed Computing, Stamford, CT: Gartner Group, 1995.
- [4] Apple, IBM, Oracle, Sun, Netscape, Network Computer Reference Profile, May 1996. <http://www.nc.ihost.com/>
- [5] Henckel, L., Schilling, J., Baumgart, T., Kuthan, J., Network Computing in einem offenen DV-Dienstemarkt, Proceedings for the Workshop on Java in Telecommunications, Darmstadt (D), 12 - 13 May 1997
- [6] PHP/FI, a server-side html-embedded scripting language, <http://php.iquest.net/>

# **Part Nine**

---

## **Internet Networking**



# Integrated Services: IP Networking Applications

*Graham Howard*

*Siemens Internet Solutions*

*Siemens Public Communications Networks*

*Siemens Telecom Networks*

*900 Broken Sound Parkway*

*Boca Raton FL 33487*

*U. S. A.*

*Phone: +1 (561) 955-8237*

*Fax: +1 (561) 955-6477*

*Email: [graham.howard@stn.siemens.com](mailto:graham.howard@stn.siemens.com)*

## **Keywords**

Internet, Applications, Convergence, Voice over IP, VoIP, Fax over IP, Middleware, Telecommunications,

## **Extended Abstract**

### **1. INTRODUCTION**

Many people have described the Internet and traditional telecommunications as being on a collision course, and portrayed these networks, and their respective operators and suppliers, as being mortal enemies.

In fact, these two networks are on a convergence path. With the great strengths of traditional circuit-switched, signalling system #7 based, telecommunications, and computing / IP-based data, video and voice communications merging into a new, multifunctional, location independent, world-wide, seamless communications medium.

## 2. IP TELEPHONY DRIVING FORCES – ECONOMICS NOT TECHNOLOGY

It is important to understand that IP telephony is not a new technology in search of a customer. It is a clear response to intense market, regulatory and competitive pressures, which are examined in detail. This includes a review of current industry value chains, the re-regulation of the communications industry, and the availability of new, low-cost technical solutions to these challenges. The importance of end-user acceptance as a key criterion is also examined.

The new telecommunications environment will thus consist of new value chains, new challenges and opportunities for existing carriers and new opportunities for second-generation carriers. These are also described in detail, with special emphasis on IP telephony as the key enabling technology.

## 3. IP TELEPHONY – REQUIREMENTS FOR TRUE CARRIER GRADE

Many people have confused IP Telephony with simple VoIP gateways. The paper includes a detailed review of the architecture of an IP Telephony system, and describes all the elements required for a true carrier-grade solution – which demonstrates it is far more than just providing gateways.

## 4. ADVANCED IP TELEPHONY NETWORK SERVICES

IP Telephony is thus a solution that addresses increasing competition, re-regulation and customer needs. It enables new service combinations, including telephony, data, networking, video, high-quality audio, multimedia messaging, electronic commerce, call centers, service centers, web / PSTN integration, and a myriad of other services. It is these that will drive IP telephony, not just cheaper POTS (Plain Old Telephone Service), and all are described in detail.

## 5. THE IMPORTANCE OF SS#7 AND AIN

Much of the existing world telecommunications infrastructure is based on SS#7. Bridging this world with the IP world is key to efficient network integration. The interaction of SS#7 and IP is examined, and illustrated by examples of some of the innovative services it can provide.

AIN (Advanced Intelligent Networks) is also an essential component, because it enables new and existing service providers to rapidly deploy advanced services, using the transport mechanisms, without having to embed large amounts of call-processing software in the network elements themselves.

The interaction of AIN and Web based features will provide the most innovative service combinations.

## 6. MANAGEMENT AND BILLING FOR THE NEW SERVICES

Contrary to popular belief, VoIP is not free - it includes sophisticated requirements for the support of both owned-network and wholesale / retail business models, as well as location independent services, global availability, bundled service packages etc. These requirements are described, as well as some of the business models they support.

## 7. CUSTOMERS AND SERVICES ARE THE KEYS

Who provides the best and most responsive services - and thus who builds and retains the largest customer base, is therefore the real measure of success.

In a truly global, IP based, communications market, the services can be provided from anywhere. Ownership of the network elements and network infrastructure, and provision of the services to customers, are therefore independent in this new communications environment.

## 8. INTEGRATED SERVICES: IP NETWORKING APPLICATIONS

The paper will conclude with an up-to-the-minute review of key industry trends, network plans and deployments and results of early service experience.

### **Full Paper**

A full text of this paper will be provided at the Conference.

### **BIOGRAPHY**

Graham Howard is currently Manager of Advanced Applications Implementation in the Siemens Internet Solutions Business Unit..

Previously he was responsible for New Product Development for Siemens Telecom Networks. Products included extensive work in Wireless systems, especially Personal Communications Services (PCS), knowledge based systems, Internet, SONET, Digital Loop Carrier, Fiber-in-the-Loop, and ISDN CPE.

Mr. Howard has extensive international experience in all branches of telecommunications, and was previously the Product Planning Executive for GPT in the United Kingdom, where he was also active in the IEE (Institution of Electrical Engineers), and lectured at several Universities.

# **The interaction of the TCP flow control procedure in end nodes on the proposed flow control mechanism for use in IEEE 802.3 switches**

*J. Wechta, A. Eberlein, F. Halsall*

*Department of Electrical & Electronic Engineering*

*University of Wales, Swansea, UK*

*eewechta@swansea.ac.uk*

## **Abstract**

The existing Ethernet and Token Ring based networks, known as legacy LANs, are no longer able to satisfy the constantly growing demand for more bandwidth and better quality of service. However, switching technology is a solution that seems to be able to cope with further expanding requirements. Introducing switches with new flow control schemes makes the networks faster and more efficient but, at the same time, this can cause other problems. An example is the interaction between the well-established end-to-end TCP flow control and the hop-by-hop switch flow control. The interaction between these two flow control mechanisms is too complex to be solved using theoretical analysis and hence simulation modelling has been used. The advantages and drawbacks of using both flow control schemes firstly independently and secondly together has been investigated and the results are presented and discussed in the paper.

## **Keywords**

**End-to-end, hop-by-hop, TCP, flow control, 802.3, switched LANs**

## **1 INTRODUCTION**

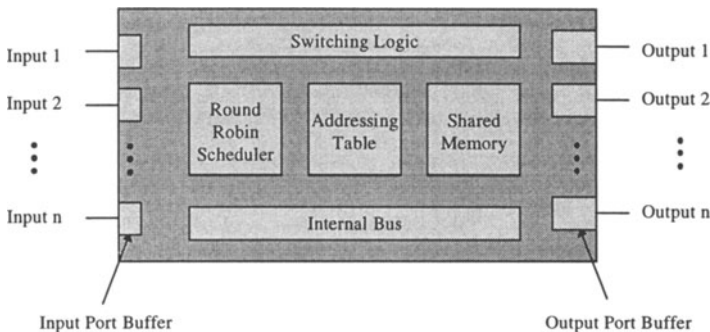
The advent of multimedia and related applications is resulting in a significant increase in the bandwidth demands on existing local area networks (LANs) [Ban91]. To meet these demands high-speed derivatives of existing LANs have been developed that operate using switching [Hal96], [Mol96], [Kun97]. Because of their origin they are referred to as IEEE 802.3 switches, a schematic diagram of which is shown in Figure 1.

As can be seen a switch operates using a fixed number of input-output ports with an internal bus architecture that enables frames to be passed from multiple input ports to multiple output ports concurrently [Hel97]. Output buffering is provided at each output port to allow for the possibility of multiple frames arriving at different input ports for the same output. However, the amount of buffering is limited [Bra97] and, when a defined limit has been reached, a flow control mechanism is used between switches to allow the congested switch to request the switches that are connected to its input ports to temporarily stop sending any new frames until the congestion has passed. This method is known as backpressure [Omi96], [Yan95]. The switch flow control mechanism is built into the input port of each switch. It uses a combination of XOFF/XON thresholds and XOFF/XON control frames to ensure that no data frames are dropped due to a lack of free memory in the switch [Wan91].

As will be expanded upon in Section 3, the flow control mechanism that has been proposed for use by switches, affects all sources and hence, as will be seen, this has an impact on the flow and congestion control procedure associated with the Transmission Control Protocol (TCP) running in each end system/host. The TCP flow control procedure [Ste95], [Com95], [Vil94] comprises two main elements: a TCP window mechanism and ACK control frames. The integrated self-clocking mechanism used by TCP allows it to regulate the rate of transfer of data packets by allocating the available bandwidth to the individual traffic sources in a controlled and predictable way.

The research reported in this paper has been carried out in order to investigate the interaction between the flow control used by switch and the TCP flow control scheme running in end-stations. The investigation has been carried out using simulation modelling. Each simulation has been run for a significant length of time in order to ensure that the results are reliable. For the research described in this paper the models of TCP version Reno and an 802.3 switching hub with its 802.3x hop-by-hop flow control have been developed in C++ as simulation modules for the BONEs Designer environment.

The motivation for our investigation is described in **Section 2**. Since our interests concentrate on the interaction between both flow control schemes it is necessary to ensure that the conditions for triggering the switch flow control are specified and satisfied. **Section 3** describes this issue. When both flow control mechanisms are



**Figure 1: Switch architecture**

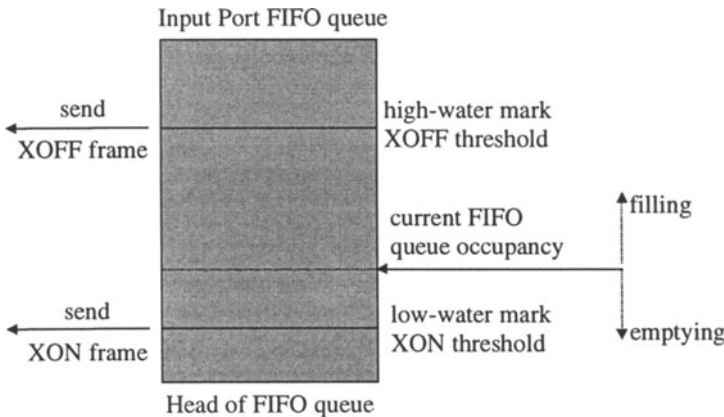
working together there is a chance that they have a negative influence on each other under certain circumstances. The simulations related to this issue and their results are described in **Section 4**. Discussion of the results obtained and conclusions drawn are presented in **Section 5**.

## 2 MOTIVATION FOR THE INVESTIGATION

The bases of the flow control mechanism used by switches is shown in Figure 2. It is a simple backpressure hop-by-hop flow control [Ger88], [Jai90], [Kun93] which is designed to cope with the effect of temporarily congested links. When frames arrive at an outgoing link faster than can be forwarded they need to be stored inside the switch and will firstly be stored within the memory inside the switch. Firstly the frames are stored in the output port buffer, then if this buffer becomes full, they are placed into the shared memory and finally, should the shared memory become full they are stored in input port(s) buffer. Hence if overload condition persist, the related input port buffers will eventually overflow [Hac96].

As shown in Figure 2 the switch flow control compares the amount of data stored in the input port buffer (FIFO queue) with the XOFF threshold value (high-water mark [Wan91]). If the XOFF threshold is reached, an XOFF control frame is sent to the adjacent switch (or the source) which in turn stops sending any new data frames. This causes the FIFO queue to release data until the current queue occupancy reaches the XON threshold (low-water mark). At this point, an XON control frame is sent to the switch/host which, in turn, allows the flow of data frames to be resumed. The levels of XON and XOFF thresholds are chosen to ensure the utilisation of affected link remains as high as possible. A problem with this type of flow control is that there is a possibility that a deadlock occurs [Ozv94].

On receipt of an XOFF message a node stops transmitting new data and waits for an XON message before resuming data transfer. If, for some reason, no XON frame arrives the node will remain silent and no more data flows on this link. The solution proposed in the IEEE 802.3x specification is to put a default delay value into the XOFF



**Figure 2: Basics of the switch flow control mechanism**

flow control frame. If a node receives an XOFF frame and does not receive XON within the delay specified in the XOFF frame it will start sending data again. The choice of a default delay is an interesting issue in itself, but is out of the scope of this paper. As a reliable transport layer protocol, the TCP is responsible for providing an error-free stream of bytes, delivered in the correct sequence [Com95]. Since the loss of data packets is possible, the TCP must perform retransmissions in order to achieve these characteristics.

In addition the TCP running in two communicating end-systems performs an end-to-end flow control on the data flow. The amount of data being sent by a TCP source is restricted by the window mechanism. At the beginning of a TCP connection, the TCP source can only send a small number of data packets. On receipt of the ACK frames for these data packets, the TCP source then increases the window size and sends more data [Ste95]. In this way the TCP window mechanism prevents a large number of data packets being initially sent into the network.

The window opening procedure is shown in Figure 3, and as can be seen, it has three phases :

- rapid increase (P1) known as “slow start”,
- slow increase (P2) known as “congestion avoidance”,
- constant; with the window fully open (P3).

Initially the window size is one, and the source can send just one data packet. On receipt of the ACK frame the window is increased to two packets and two new packets can be sent. For each received ACK frame, the TCP window is increased by one packet size and two new data packets will be sent. Hence, this phase is called the exponential growth zone [Ste95] since the value of the TCP window size increases exponentially [Jac88]: 1, 2, 4, 8, 16 and so on.

When the TCP window reaches half of its maximum size, the slow increase phase is entered. In this phase the growth slows down and increases by only one data packet per round trip time (RTT); that is when all data packets from the current window become acknowledged. This phase ends when the TCP window becomes fully open.

When the TCP window is fully open the TCP connection is in a kind of equilibrium, since as a packet leaves the network a new packet enters. The TCP has a self-clocking

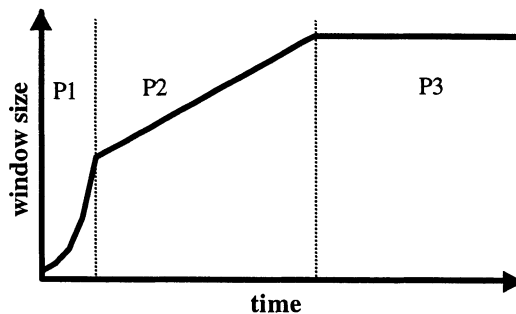


Figure 3: TCP window opening procedure

mechanism which is maintained by ACK frames. The delay experienced by a data packet and its ACK frame is used to dynamically adjust the data flow to both the available bandwidth and the prevailing network loading.

When a TCP data packet is lost, the sender can recover using either the Fast Retransmit scheme [Ste97] or to wait for the retransmission time-out to expire. The Fast Retransmit scheme is triggered on the arrival of at least three duplicate ACK frames which indicates the need for the missing packet to be retransmitted. This retransmission is followed by the Congestion Avoidance procedure [Ste97] which reduces the TCP window size by half. In contrast, if the sender relies on the retransmission time-out, the packet transmission rate is reduced drastically since the retransmission is followed by the Slow Start [Ste97] procedure with the window size reset to 1 packet. As can be seen, in both cases the value of the traffic offered by a sender to the system decreases but, in the second case, the decrease is more drastic.

In this paper the end-to-end TCP and back-pressure hop-by-hop switch control schemes have been treated as a complementary solution. However in many papers (e.g. [Mis92],[Mis96],[Ozv94]) the two flow control schemes have been examined as alternative solutions. The primary motivation for carrying out the research therefore was to investigate the effect of operating both flow control schemes concurrently. To highlight any effects, various combinations of the two flow control schemes being enabled or turned off have been investigated.

### 3 THE ONSET OF SWITCH FLOW CONTROL

As indicated earlier, there are three types of buffer within a switch: the output port buffer, the shared memory and the input port buffer. This is shown in diagrammatic form in Figure 4. The input port buffer is a buffer dedicated to the end-node connected to this input port and hence has exclusive use of this buffer. The primary task of an input port buffer is to hold received packets awaiting transfer to an output port during periods when short bursts of data packets arrive. The output port buffers are provided to hold packets during the times when the output link is heavily loaded. The shared memory provides additional storage which can be used to hold waiting packets, when destination output port buffer is full, and increases the flexibility of the internal structure of the switch. If there is a congested link on the path between a TCP source and a TCP receiver, the TCP data segments will not be relayed but instead stored within the affected switch. These data segments may be kept in the available memory

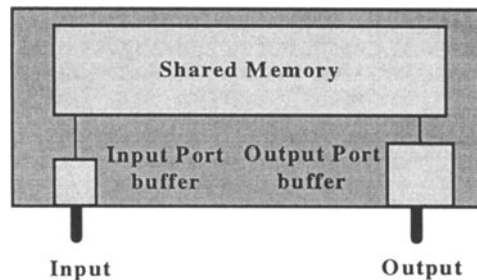


Figure 4: The switch memory storage components



storage in each switch along the path [Bol93]. However, the available storage will not be filled randomly. The storage which is the closest to the congested link will be filled first, then the next closest and so on. Hence, the sequence of the storage is filled inside each switch is as follows: first the output port buffer is filled, then the shared memory, and finally the input port buffer. To illustrate the effect of congestion, we assume the three-stage switch topology as shown in Figure 5.

It should be stressed that XOFF/XON traffic will only be present on an input link of a switch that has not enough memory left to store the incoming data packets. Since the TCP window mechanism regulates the amount of data currently in transit, it is helpful to first analyse in more detail the amount of data which is produced by TCP sources, and the available storage in the switches along the path. In practice this was done before the main set of simulations related to flow control interaction were carried out.

TCP restricts the maximum amount of data in transit with the help of the TCP output window. Hence it is possible to determine the maximum overall amount of data currently in transit. Although the maximum TCP window size is usually 64 kB [Ste95], a maximum window size of 128 kB has been chosen for this simulation. Such a large TCP window size was necessary in this simulation in order to fill the switch storage and to trigger the switch flow control. The same simulation results can be obtained by using a larger number of data connections with smaller TCP window size (e.g. 64kB or 32kB) or by using a smaller TCP window size and less storage within the switch. In order to keep the simulation time low the configuration with 10 clients and 128kB window size has been chosen. Since the window size is anticipated to be larger in the future, the TCP header allows the definition of the Window Size Option which supports window sizes up to 1 GB.

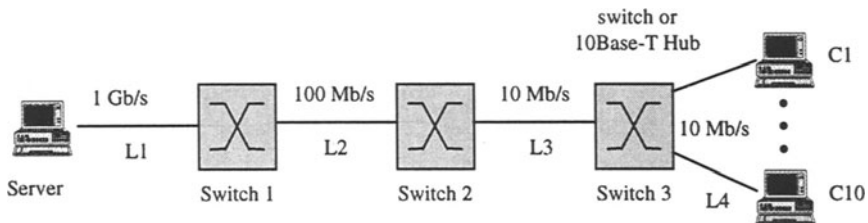
We assume that all clients request file transfers from the server. Each client runs a receiver process and establishes a connection to a sending process running on the server. We examined the worst possible case where all data transfers originated at more or less the same time and flowed from server to clients. The following equation describes the relation between the amount of data in transit, the number of TCP sender processes (later on referred to as a source) and the maximum window size:

$$A = N * \text{MaxTCPWindowSize} \quad (1)$$

The following parameter values were used in all equations:

$N$  = number of connected TCP sources,  $\text{MaxTCPWindowSize} = 128\text{kB}$ ,

$\text{OutputPortBufferSize} = 32\text{kB}$ ,  $\text{InputPortBufferSize} = 16\text{kB}$ ,  $\text{SharedMemorySize} = 512\text{kB}$



**Figure 5: The examined three-switch topology**

The value of A is shown in Figure 6, which indicates the maximum amount of data which is currently in transit when the TCP output window of each source is fully open, that is, 128kB. The points below line A can be interpreted as the possible amount of data in transit if the TCP windows of the sources are not yet fully open. Under congestion conditions the buffers of switch 2 are filled first. The amount of data that may be stored inside Switch 2 is shown in Figure 5 using the equation of line B. The equation for line B is as follows:

$$B = \text{OutputPortBufferSize} + \text{InputPortBufferSize} + \text{SharedMemorySize} \quad (2)$$

Since the equation is not a function of the number of TCP sources, line B in Figure 6 has the constant value of 560kB.

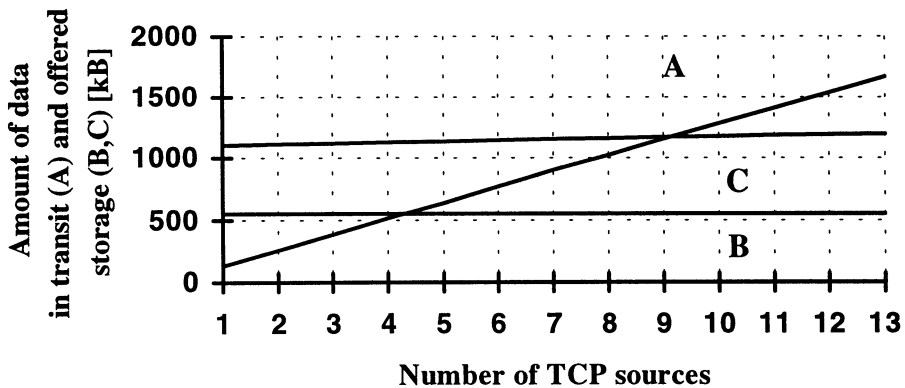
The size of the input port buffer is 16kB. However because of the XOFF/XON thresholds, the occupied space of the Input buffer varies. It can be estimated as the average value of the two extremes, when the input port buffer is full (16kB) and when it is empty (0kB). The average value is thus 8kB. Nevertheless, in order to provide a worst case calculation, the value of 16kB has been chosen, which, in practice, will only occur in rare situations.

If link L3 continues to be congested, the memory buffers in the preceding switches will start to be filled. It should be stressed that the buffers of switch 1 will only be filled when all the buffers of switch 2 have already been filled. Hence the combined memory available is as shown in Figure 6. The equation for line C is as follows:

$$C = B + \text{OutputPortBufferSize} + N * \text{InputPortBufferSize} + \text{SharedMemorySize} \quad (3)$$

Where: B = storage inside switch 2

As can be seen in Figure 6, the amount of memory available (Line C) increases with the number of TCP sources in the system. In summary, therefore, line A represents the amount of data in transit when  $N$  TCP sources are sending frames. Line B represents the storage available inside switch 2 and line C represents the amount of storage inside both switches 2 and 1.



**Figure 6: Amount of data in transit between multiple TCP source-receiver pairs and the amount of offered storage**

Lines B and C divide the amount of data in transit (traffic offered) into three areas:

- below B: traffic offered is smaller than the capacity of the storage of switch 2;
- above B and below C: traffic offered exceeds the storage capacity inside switch 2 but it is still lower than the combined storage capacity of switches 2 and 1;
- above C: traffic offered exceeds the storage capacity of both switches.

The part of line A below line B relates to a topology with 1, 2, 3 or 4 TCP sources, which means that the amount of data produced by them is not sufficient to fill the storage inside switch 2.

The part of line A between lines B and C relates to a topology with 5, 6, 7, 8 or 9 TCP sources. This means that in order to prevent the loss of TCP data packets, XOFF frames need to be generated by the input port of switch 2 and sent to switch 1, that is., XOFF/XON traffic becomes present on link L2 between switches 2 and 1.

The part of line A above line C relates to a topology with 10 or more TCP sources. In this case there is enough traffic offered to fill the storage inside both switch 2 as well as inside switch 1. When the storage inside switch 2 becomes full, XOFF frames will be generated at the input port of this switch and sent to switch 1. When switch 1 receives XOFF frames, it will start to fill its own buffers. When these also become full, all input ports of switch 1 will generate XOFF frames.

Having produced the above graph we are now able to predict when XOFF/XON traffic will be present on any particular link. Furthermore, we can do similar predictions for different parameters such as the amount of shared memory, output port buffer, input port buffer size or the TCP window size. In this case we need to recalculate the equations and derive new graphs. Similar graphs can be derived for different topologies and with varying numbers of switches.

#### 4 SIMULATION RESULTS

The first set of simulations was carried out in order to investigate the effect of enabling and disabling the switch flow control on the throughput and the loss of data packets within the switches. These were carried out for the combined three-switch topology shown earlier in Figure 5. Four different cases were investigated. Some allow for packets being dropped when the storage capacity within a switch is exceeded (switch flow control disabled on selected links) which gives an insight into the TCP retransmission scheme. Others allow us to compare these results with the results of the lossless configurations when the switch flow control is enabled on all links and no data packet loss is expected.

As can be seen in Figure 5, the topology comprises three switches and  $N$  TCP sender process / receiver process pairs. The bandwidth of link L1 is 1Gb/s, of link L2 it is 100Mb/s, while the bandwidth of links L3 and L4 is 10Mb/s. The application for all senders is assumed to be the transmission of an infinite file using FTP. The maximum TCP window size is set to the value of 128kB, all links have the same length, the size of the data packets is constant.

The following four cases have been examined:

- 1) switch flow control enabled on all links (i.e., end-node-to-switch links and on switch-to-switch links),
- 2) switch flow control enabled on switch-to-switch links only,
- 3) switch flow control enabled on end-node-to-switch links only,
- 4) switch flow control disabled on all links.

During the investigation the most important parameters quantified were the **throughput** on each link, the **bandwidth** used by a particular sender-receiver pair, and the **number of lost packets**.

#### 4.1 Switch flow control enabled on all links

The *TCP window graphs* in Figure 7(a) show how the size of the TCP window in each end-node changes with time. The graphs show the continuous growth of the size of the TCP windows of 10 sender processes within the server. The growth is over when the TCP window is fully open (128kB). No deviations occur on the graphs which means that no packet retransmissions have been performed by any of the TCP traffic sources. The three phases of the TCP opening procedure (rapid increase, slow increase, and no increase) can be clearly seen.

The *Shared Memory Occupancy graphs* in Figure 7(b) show the rate at which the shared memory is filled in the three switches. The graph contains two curves showing the process of filling the storage inside switch 2 and in switch 1. It can be seen that the storage inside switch 2 is filled quickly. This phenomenon should be seen as a response to the rapid growth of the TCP window and the large number of data packets injected into the system on all links with a bitrate of 100 Mb/s, but the buffers in switch 2 are emptied slower with a bitrate of 10 Mb/s. The process of filling the storage inside switch 1 takes a much longer time (about 20s). This can be explained by each TCP source entering the phase of slow growth of the TCP window and hence a smaller rate of new data packets being injected into the system. No variations in the curves were registered which means that the flow of incoming data packets was constant. Switch 3 is ahead of the bottleneck link L3 and all packets arriving at this switch are immediately relayed to the end-nodes. Therefore no packets are stored in the shared memory within switch 3 and the shared memory occupancy curve for this switch is shown to be 0kB. All graphs show that the system is in a steady state after approximately 30s of simulation time.

#### 4.2 Switch flow control enabled on switch-to-switch links only

The *TCP window size graphs* shown in Figure 8(a) show that the TCP senders experience varying levels of throughput when flow control on the switch-to-end-node links is disabled. This is illustrated by the different times when the window sizes of the senders drop. As can be seen, the ten nodes are divided into the three groups:

- The senders 3, 5 and 8 do not experience a drop of window size. This means that the data packets from them are transmitted first over the slow link resulting in the related ACK frames being received and their corresponding windows carry on increasing.

Scale=10↑3

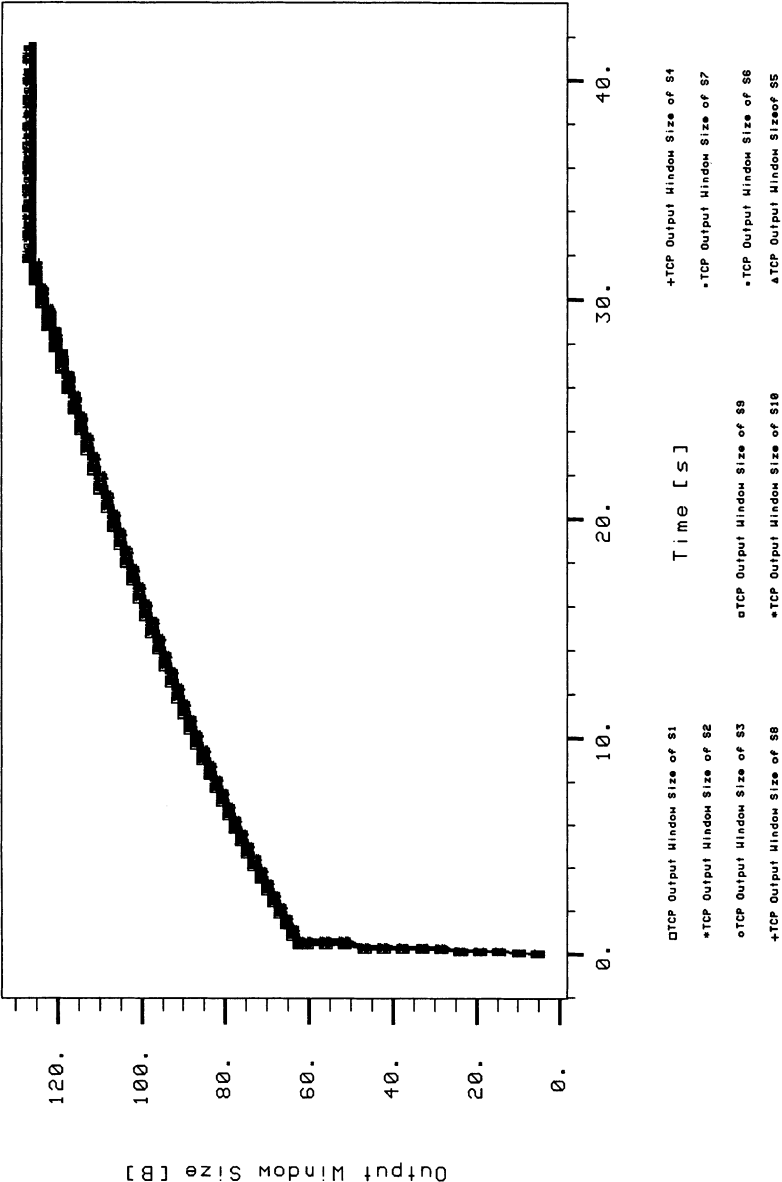


Figure 7(a): The variation of the TCP window size with time (FC enabled on all links)

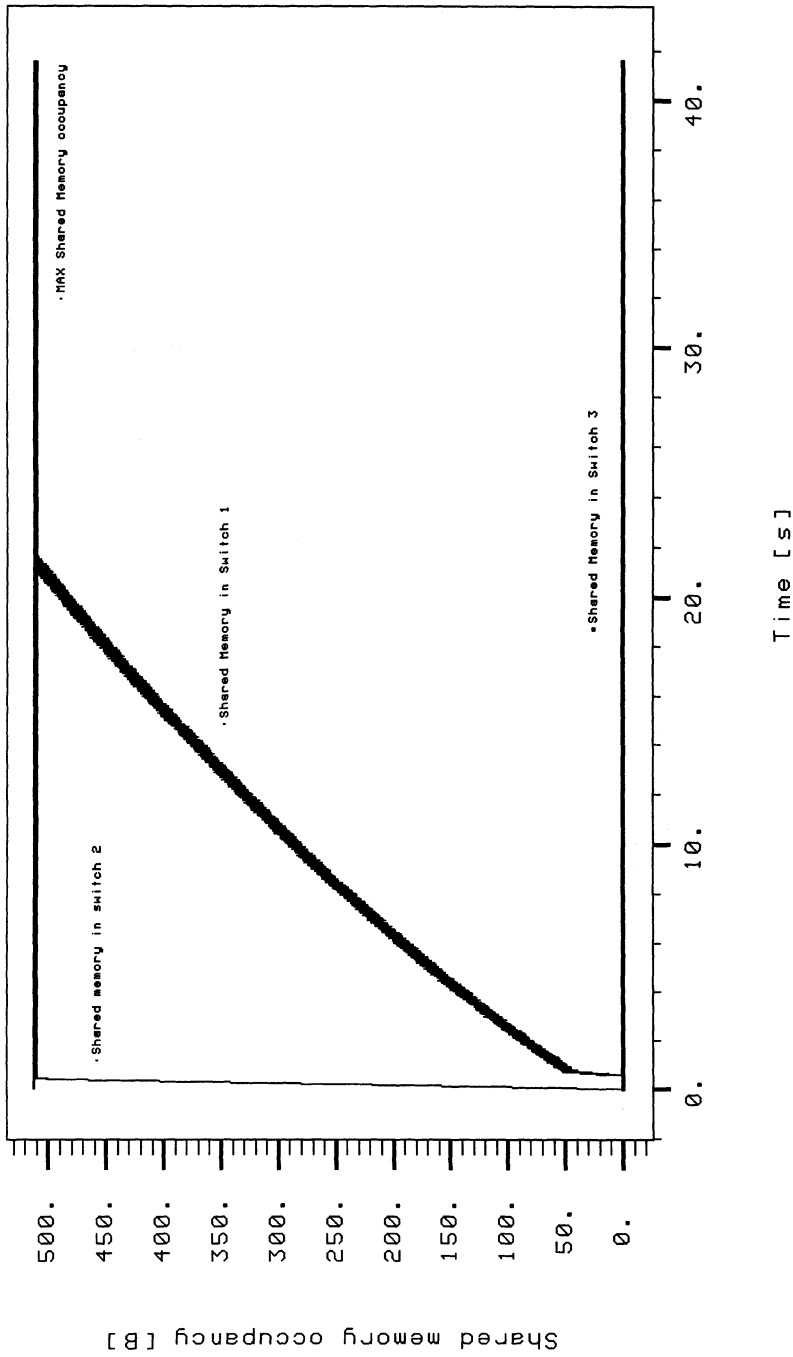
Scale=10<sup>3</sup>

Figure 7(b): The variation of the Shared Memory Occupancy with time (FC enabled on all links)

Indeed, once the data packets from the other sources stop being transmitted, there is an increase in the rate at which these windows increase.

- The senders 1, 2, 6 and 7 have each experienced one retransmission and as a result their window sizes have been divided by two ( retransmissions are the result of data packets being discarded in switch 1)
- The senders 4, 9 and 10 have each experienced two retransmissions and as a result their window sizes have been divided twice by two.

The sender numbers have only been assigned in order to relate to the particular senders later on. It needs to be stressed that the packets are dropped at random.

*The Shared Memory Occupancy graphs* in Figure 8(b) show the effect of data packet retransmissions on the buffer occupancy in the switches. Since the input flow has been slowed down by TCP, but the output remained unchanged, the number of data packets starts to fall. This can be explained by treating the storage inside switch 1 as an extension of the memory inside switch 2. If the line showing the shared memory occupancy inside switch 1 is put above the line showing the memory occupancy inside switch 2, then the line showing the maximum occupancy becomes the minimum occupancy line of switch 1. It can thus be deduced that the whole storage of switch 2 has been emptied and 100KB of data from the storage of switch 1 has been sent before the recovery starts and incoming data starts to fill up the storage again.

The number of retransmitted packets per sender were counted during the simulation:

- senders 3, 5 and 8 experienced no retransmission,
- senders 1, 2, 6 and 7 had one retransmission, and
- senders 4, 9 and 10 have had two retransmissions.

In this case, when switch flow control is enabled on switch-to-switch links only, all TCP output windows were already in the slow increase phase when packets started to get lost. So the effect on a particular sender was relatively small. This is best discussed in relation to the topology which is shown in Figure 9. The storage in switches 2 and 1 were already filled up when the switch flow control was supposed to appear on the switch-to-end node links. However, since the switch flow control was disabled, it was not triggered and hence data packets were lost.

Figure 9 shows the scenario where flow control is enabled on the switch-to-switch links, however disabled on the switch-to-end node links. This means, that after filling the storage in switches 2 and 1, there is no means to send a message to the end node's MAC interface with the request to stop transmitting new data packets. The crosses show the places where data packets are being dropped. The switches in dark grey those switches in which the storage is full before packets are dropped.

### 4.3 Switch flow control enabled on switch-to-end-node links only

Figure 11 shows the scenario where the switch-to-switch flow control is disabled. After filling the storage in switch 2, it has no means to send a message to switch 1 to request it to stop transmitting new data packets. The cross shows the place where data packets are being dropped. The TCP senders produced only as much data as it was necessary to fill the storage inside switch 2 which means that the senders are still in the phase of

Scale=10↑3

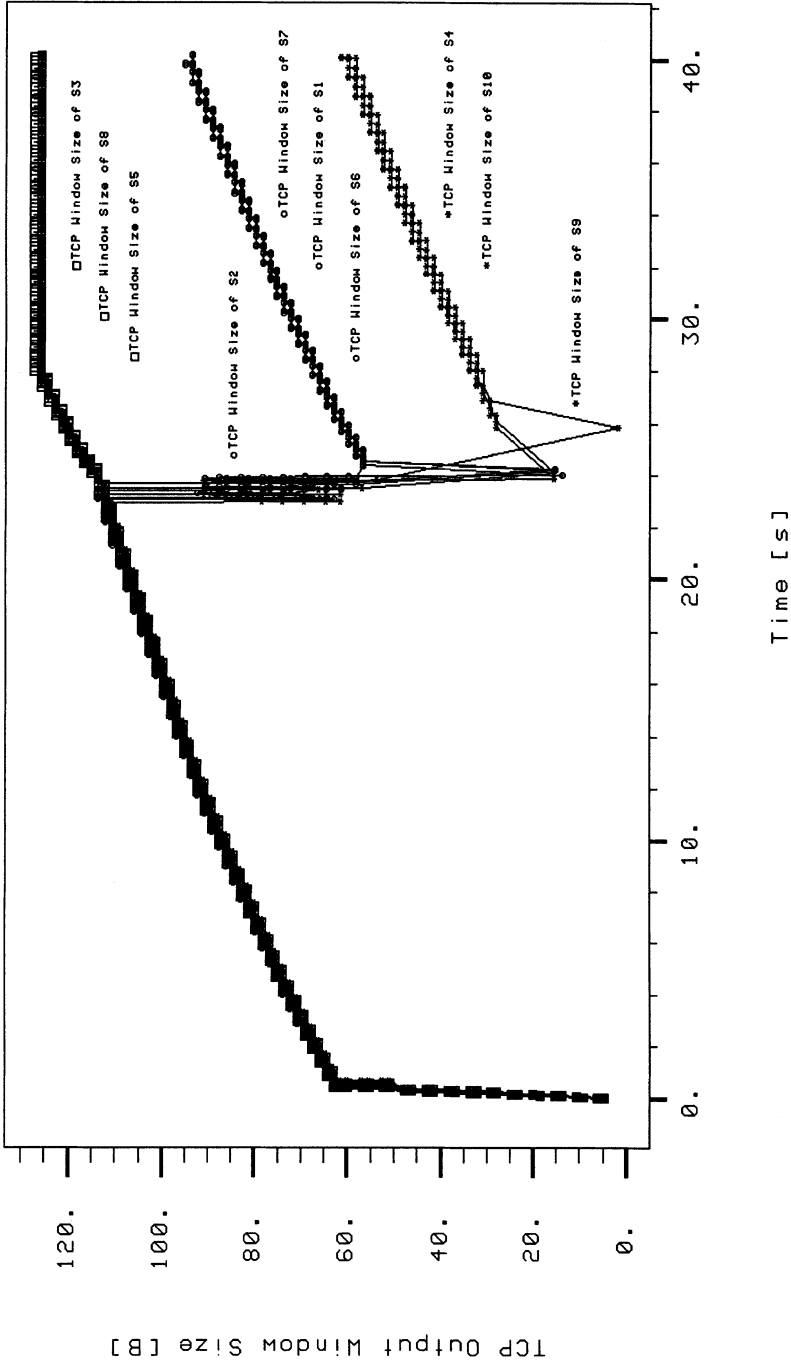


Figure 8(a): The variation of the TCP window size with time (FC enabled on switch-to-switch links only)



Scale=10↑3

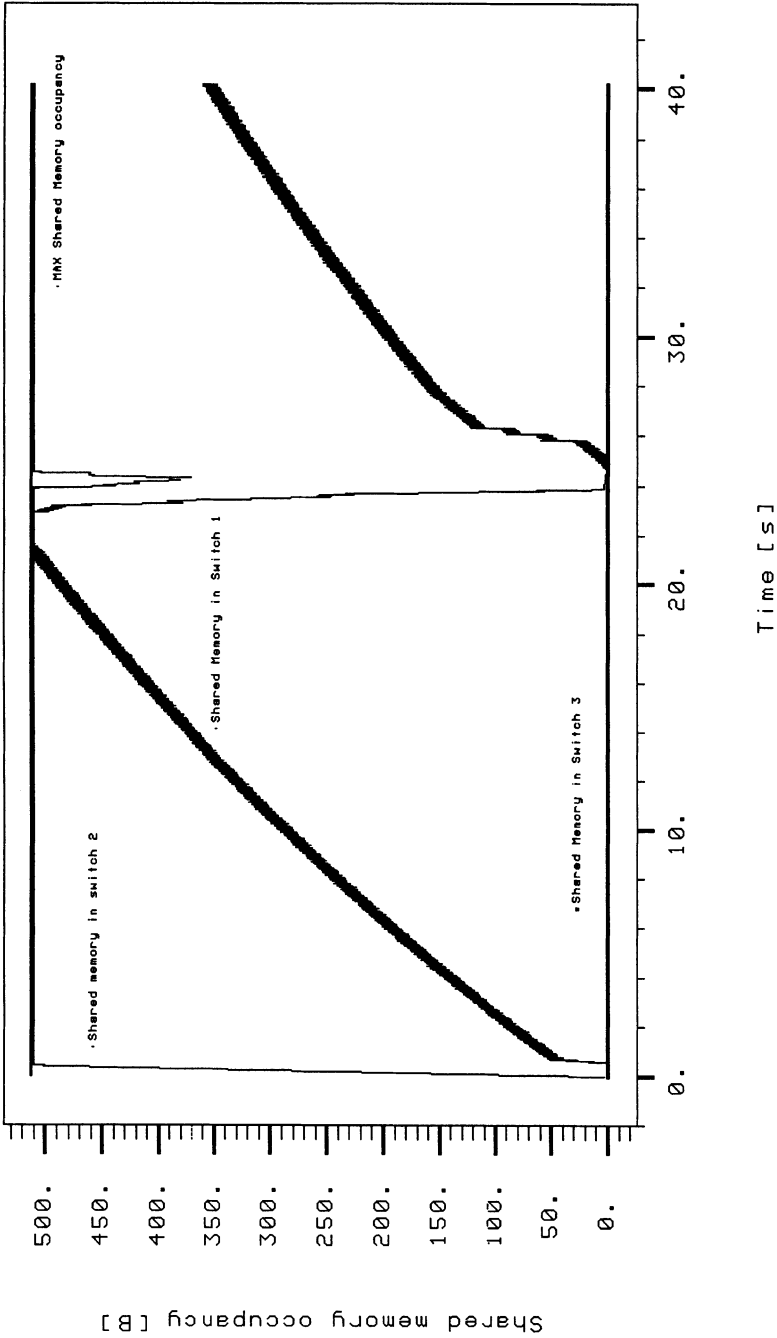
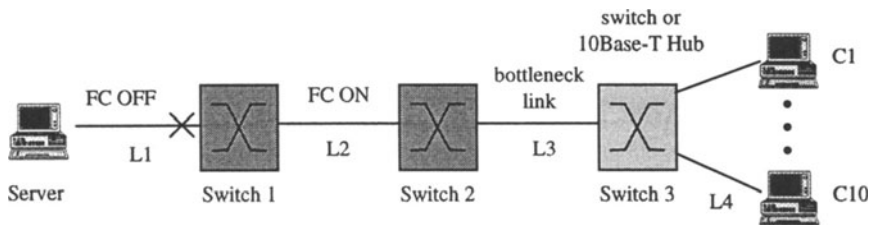


Figure 8(b): The variation of the Shared Memory Occupancy with time (FC enabled on switch-to-switch links only)



**Figure 9: Flow control disabled on links connecting senders and switch 1**

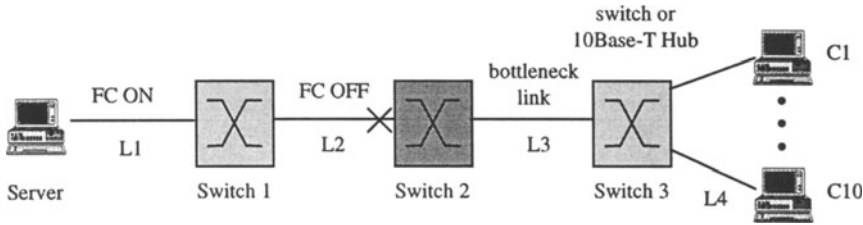
rapid increase of the TCP window size. The switch in dark grey indicates that switch in which the storage is full before packets are dropped.

The *TCP window size graphs* in Figure 10(a) show that all TCP senders are affected by packet losses. In the steady state, all the TCP output window sizes are divided by two. Initially, however, in the case of senders 1, 2, 3 and 4, recovery starts after the initial fall. In the case of sender 5 a second retransmission is performed. For the remaining senders the situation becomes even worse because of the loss of data packets and the need to rely on the retransmission time-out to expire (senders 7, 8, 9 and 10).

The interesting point, however, is that in the steady state all TCP senders, regain the fairness in sharing the bandwidth. The effect of this is a similar window size for different senders even if large variations occurred during the earlier stage. It can also be seen that when the sum of all TCP sender window sizes (the amount of data currently in transit) exceeds the size of storage inside switch 2, packet losses and hence further window size reductions are unavoidable.

The *Shared Memory Occupancy graphs* in Figure 10(b) show some regularity between the emptying and filling of shared memory buffers. However, what needs to be remembered is that each time the storage within the switches becomes full, all new incoming data packets are rejected.

A TCP sender which experiences data packet losses will retransmit the affected data packets. In the examined topologies the highest number of data packet retransmissions occurred in the case when the switch flow control is disabled on the switch-to-switch links. This can be explained by the phase of the TCP window opening procedure. The TCP windows were in the rapid increase phase when packet drops started to occur after filling up the storage in switch 2. Since the flow control on the switch-to-switch links was disabled, incoming data packets were dropped. During the rapid increase of the TCP window more data packets are injected into the system in one burst (2 packets per



**Figure 11: Switch flow control disabled on switch-to-switch link between switches 1 and 2**

Scale=10<sup>13</sup>

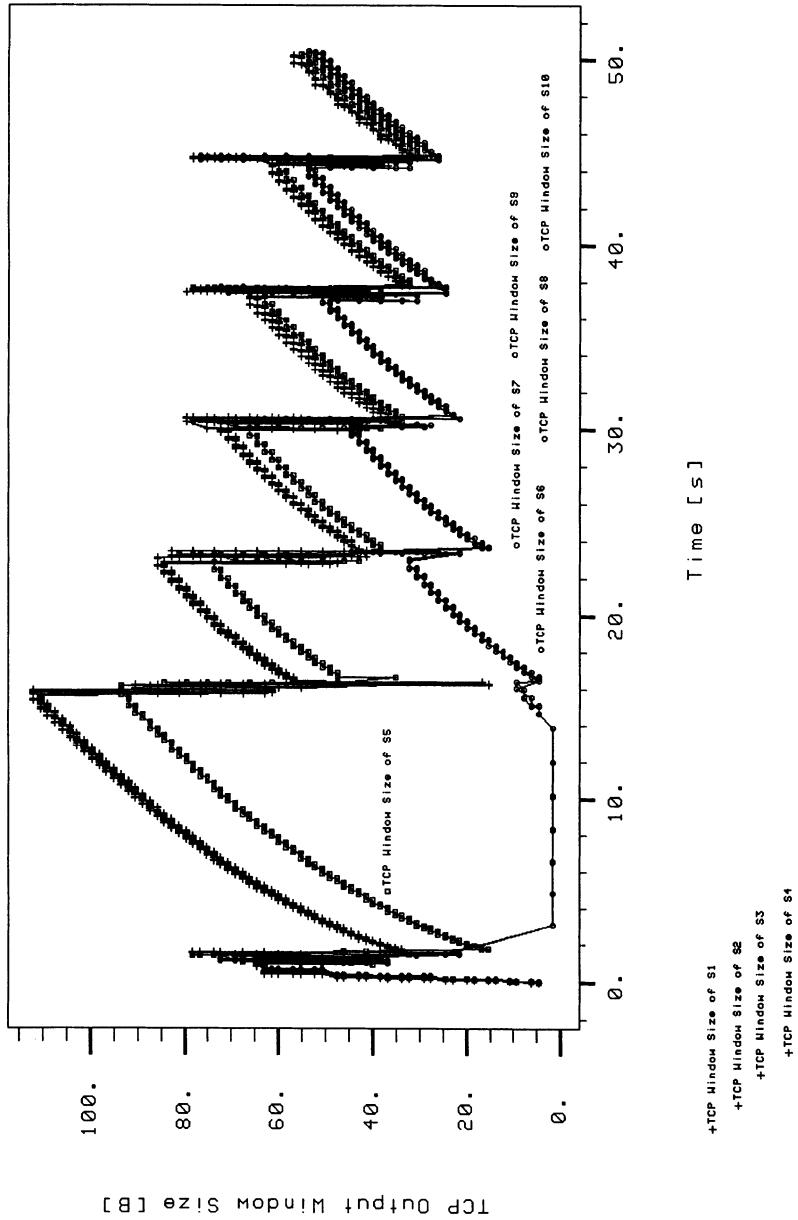


Figure 10(a): The variation of the TCP window size with time (FC enabled on switch-to-end-node links only)

Scale=10<sup>3</sup>

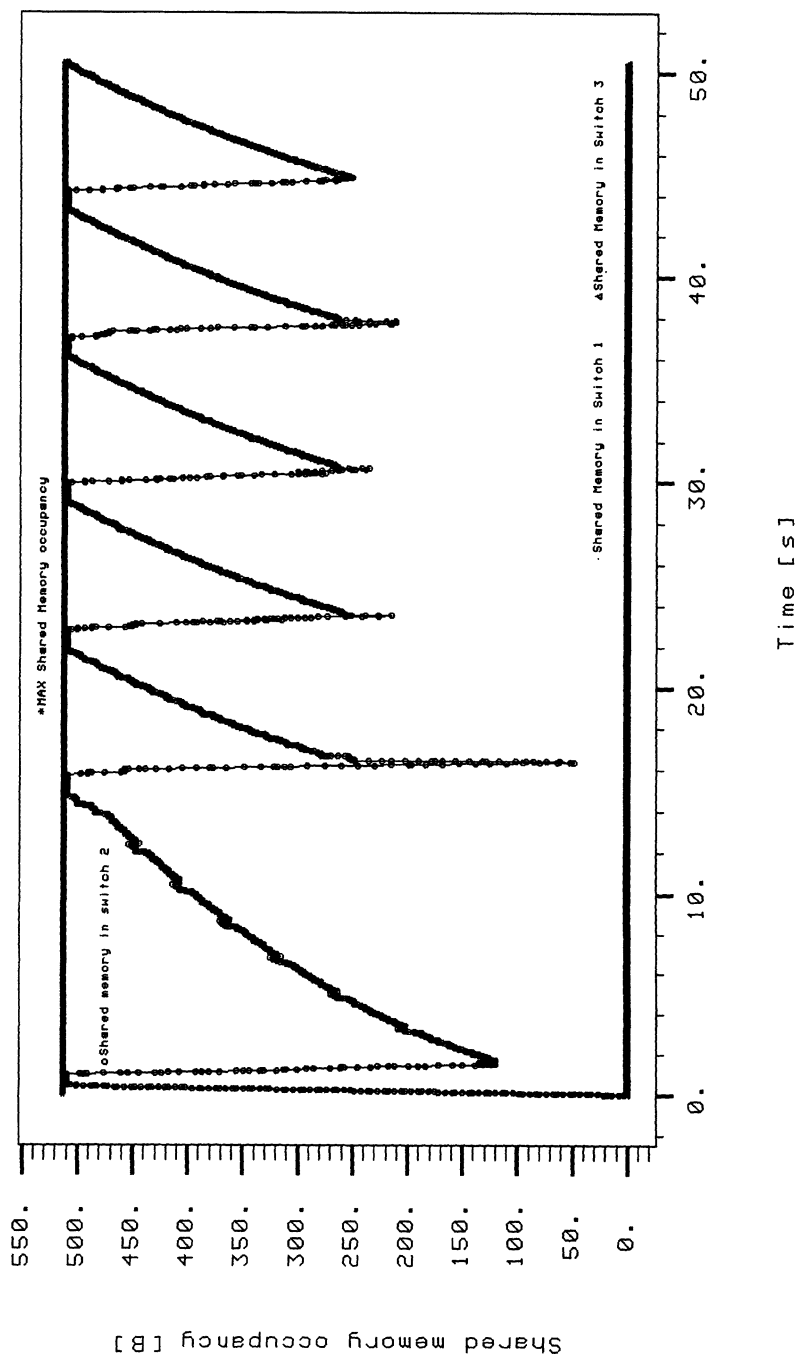


Figure 10(b): The variation of the Shared Memory Occupancy with time (FC enabled on switch-to-end-node links only)

ACK) which means that the probability of data packet losses from the same source is much higher.

#### 4.4 Switch flow control disabled on all links

This case yields exactly the same results as the case where switch flow control is enabled on the switch-to-end-node links only and disabled on the switch-to-switch links. As just observed, if the switch-to-switch link is run without switch flow control new incoming data packets will be dropped when the storage in switch 2 becomes full. Without the XOFF/XON flow control frames, switch 1 will not be aware of storage problems experienced by switch 2. Hence switch 1 will always forward incoming data packets towards switch 2, even if the latter is not able to store them. Therefore, there is no way to fill the storage in switch 1 and hence the XOFF/XON flow control will not be triggered on the links to the end nodes.

### 5 DISCUSSION AND CONCLUSIONS

The network resources comprise two components: link bandwidth and buffer capacity within network nodes, e.g., switches. The switch flow control procedure copes with problems related to packets that are already in transit across the network and supports the data packet flow on a hop-by-hop basis. The TCP flow control decides when and how much new data may be injected into the network. Since the TCP is an end-to-end flow control protocol it can be considered as operating on a higher level than the examined switch flow control operating on a hop-by-hop basis.

A TCP traffic source sees the whole set of switches in a route as a long fat pipe. Hence, it does not know that there is a string of switches in-between and it is not aware of any possible problems, which can emerge when, for example, a storage overflow occurs within switches. Clearly, TCP does not have any mechanism which can prevent data packets being dropped due to buffer overflow within switches. The TCP needs a flow control manager on a hop-by-hop level, and the switch flow control performs this role. Concluding, the TCP governs the process of bandwidth sharing, while the switch flow control performs the management of storage within switches. Hence the TCP and switch flow controls operate in different areas and complement each other very well, at least for the analysed topology. However, it is possible to find network topologies and node configurations where using the hop-by-hop flow control can significantly reduce the performance of a network. This phenomenon is described in detail in [Wec98].

Regarding the system response, the TCP flow control is slow, while the switch flow control is fast. There is a delay before the TCP reacts to any indication of congestion. The TCP flow control procedure is based on the timing mechanism maintained by the ACK frames. Since the delay in the examined topology is a function of the number of packets in transit or/and stored in buffers, it is very dynamic and much longer than the packet transmission time over one link. The delay related to switch activities is equal to the transmission delay on a link between two neighbouring switches. Clearly, this is a very short time in comparison to the RTT. However, the actions taken by switches to ensure the smooth flow of data packets through the system remain transparent to the TCP.

Simulation results show that TCP traffic sources, thanks to the window opening procedure, produce much more data than could be transmitted over the network, and hence this data was stored within switches. It should be noted that there is a common belief that long queues are a symptom of congestion. Since such extensive queuing has a negative influence on the end-to-end delay as well as increases the probability of packet losses when new flows are being admitted, one can ask the question whether this effect can be avoided.

The results for the system without switch flow control show that the data flow synchronisation effect [Flo93] is a result of packets being fairly dropped from all flows due to buffer overflow. This is a very harmful effect because it can cause strong load fluctuations in the network and jumping between two temporary states: overloading and underutilising of network resources. Solutions to this problem have been proposed in [Flo93], [Flo97] and can be implemented by introducing a new scheme for dropping packets in congested nodes on a random basis rather than using a simple drop-tail approach. However, combining end-to-end and switch flow control schemes avoids packet dropping since it prevents buffer overflow.

On the other hand dropping packets is considered a proper response from the network but data packet retransmission means wasting valuable bandwidth. Any possible solution which does not waste bandwidth is worth investigating.

## 6 REFERENCES

- [Ban91] Bandula, W. (1991) High-Speed Local Area Networks and Their Performance: A Survey, *ACM Computing Surveys*, **23**(2), 221-264.
- [Bol93] Bolot, J. (1993) End-to-End Packet Delay and Loss Behaviour in the Internet, *Proceedings ACM SIGCOMM'93*.
- [Bra97] Braden, B., Clark, D. and Crowcroft, J. (1997) *Recommendations on Queue Management and Congestion Avoidance in the Internet*, Internet draft: draft-irtf-e2e-queue-mgt-00.ps.
- [Com95] Comer, D. (1995) *Internetworking with TCP/IP Vol. 1, Principles, Protocols and Architecture*, Prentice Hall.
- [Els96] Elsaadany, A., Singhal, M. and Liu, M.T. (1996) Performance Study of Buffering within Switches in LANs, *Computer Communication*, **19**, 659-667.
- [Flo93] Floyd, S. (1993) Random Early Detection Gateways for Congestion Avoidance, *IEEE/ACM Transactions on Networking*, **1**(4), 397-413.
- [Flo97] Floyd, S. (1997) *Router Mechanisms to Support End-to-End Congestion Control*, Technical Report, Lawrence Berkeley National Laboratory, <http://ftp.ee.lbl.gov/floyd/red.html>.
- [Ger88] Gerla, M. and Kleinrock, L. (1988) Congestion Control in Interconnected LANs, *IEEE Network*, **2**(1), 72-75.
- [Hac96] Hac, A. (1996) Bandwidth Management and Switch Buffer Allocation in High-Speed Networks with Bursty Traffic, *International Journal of Network Management*, **11**(4), 2-16.

- [Hal96] Halsall, F. (1996) *Data Communications, Computer Networks and Open Systems*, Addison Wesley.
- [Hel97] Held, G. (1997) *High-Speed Networking with LAN Switches*, Wiley Computer Publishing.
- [Jac88] Jacobson, V. (1988) Congestion Avoidance and Control, *Proceedings of ACM Sigcomm'88*, **18**(4), 314-329.
- [Jai90] Jain, R. (1990) Congestion Control in Computer Networks: Issues and Trends, *IEEE Network Magazine*, 24-30, May 1990.
- [Kun93] Kung, H., Morris, R., Charuhas, T. and Lin, D. (1993) Use of Link-by-Link Flow Control in Maximizing ATM Network Performance Simulation Results, *Proceedings of IEEE Hot Interconnects Symposium*, 1-12.
- [Kun97] Kung, H. (1997) *Traffic Management for High-Speed Networks*, Fourth Lecture, International Science Lecture Series, National Academy of Sciences.
- [Mis92] Mishra, P. and Kanakia, H. (1992) A Hop by Hop Rate-Based Congestion Control Scheme, *Proceedings of ACM Sigcomm'92*, 112-123.
- [Mis96] Mishra, P., Kanakia, H. and Tripathi, S. (1996) On Hop-by-Hop Rate Based Congestion Control, *IEEE ACM Transactions on Networking*, **4**(2), 224-239.
- [Mol96] Molle, M. (1996) 100Base-T/IEEE 802.12/ Packet Switching, *IEEE Communications Magazine*, pp. 64-73. August 1996.
- [Omi96] Omidyar, C. and Pujolle, G. (1996) Introduction to Flow and Congestion Control, *IEEE Communication Magazine*, 30-32, November 1996
- [Ozv94] Ozveren, C., Simcoe, R. and Varghese, G. (1994) Reliable and Efficient Hop-by-Hop Flow Control, *Proceedings of ACM Sigcomm'94*, **24**(4), 89-100.
- [Ste95] Stevens, R. (1995) *TCP/IP Illustrated, Vol. 1, The Protocols*, Addison Wesley.
- [Ste97] Stevens, W. (1997) *TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery*, RFC 2001.
- [Vil94] Villiamizar, C. (1994) High Performance TCP in ANSNET, *Computer Communication Review*, **24**(5), 45-60.
- [Wan91] Wang, Y. and Sengupta, B. (1991) Performance Analysis of a Feedback Congestion Control Policy under Non-Negligible Propagation Delay, *Proceedings of ACM Sigcomm '91*, 49-57.
- [Wec98] Wechta, J., Eberlein, A., Halsall, F. and Spratt, M. (1998) Simulation-based Analysis of the Interaction of End-to-End and Hop-by-Hop Flow Control Schemes in Packet Switching LANs, *Proceedings of the Fifteenth UK Teletraffic Symposium on Performance Engineering in Information Systems*.
- [Yan95] Yang, C.-Q. and Reddy, A. (1995) A Taxonomy for Congestion Control Algorithms in Packet Switching Networks, *IEEE Network*, 34-45, July/August 1995.

# On End-to-End Congestion Avoidance for TCP/IP

Jim Martin  
IBM Corporation  
PO Box 12195  
RTP, NC, USA 27709  
919 254 4447  
jjm2 @ us.ibm.com

Arne Nilsson  
North Carolina State University  
Box 7914, Raleigh, NC 27695  
Raleigh, NC 27695  
919 515 5130  
nilsson @ ncsu.edu

## Abstract

A TCP/IP network utilizes several congestion control schemes: end-to-end flow control and congestion avoidance, gateway congestion control, and explicit closed-loop feedback (i.e., source quench). The evolution of TCP/IP includes enhanced gateway congestion control algorithms (i.e., Random Early Detect) and a variety of incremental improvements to TCP including selective acknowledgement and possibly end-to-end congestion avoidance (i.e., TCP/Vegas). We focus on end-to-end congestion avoidance algorithms for TCP, specifically those algorithms that use change in packet transit times as an indicator of network congestion. TCP/Vegas is the most well known algorithm based on this form of congestion control. We find that TCP/Vegas does increase throughput primarily by avoiding time-outs. However its assessment of congestion is prone to significant error which can lead to increased queue levels at the bottleneck link. By studying TCP/Vegas and other algorithms, our goal is to understand the issues associated with end-to-end congestion avoidance schemes that monitor change in packet delays.

This paper is organized as follows. First we introduce end-to-end congestion avoidance. Next, using simulation, we explore the various issues and challenges associated with end-to-end congestion avoidance by demonstrating and analyzing several end-to-end congestion avoidance algorithms. We conclude with a discussion of key issues associated with end-to-end congestion avoidance and identify future work.

## Keywords

Congestion control, TCP, TCP/Vegas, congestion avoidance



## 1 INTRODUCTION

A congestion control scheme can be classified as either reactive or preventive (the latter is also known as congestion avoidance). Additionally, some control schemes require feedback while others do not. A reactive scheme inherently is closed-loop while preventive schemes can be either open or closed-loop. Open-loop control is inherently preventive, employing admittance control and/or traffic policing to prevent congestion from occurring. Closed-loop congestion avoidance, on the other hand, is designed to keep the network at the point of maximum power (i.e., the point where the ratio of throughput versus delay is highest).

The feedback in a closed-loop system is either implicit or explicit. Explicit feedback involves an explicit send of feedback information. Explicit feedback can be characterized by the location of the source of the feedback (i.e., referred to as the level of control), by the mechanism that transfers the feedback to the source and by the actual content of the feedback. Various forms of explicit feedback exist in the Internet today such as TCP's end-to-end flow control [9], source quench [11] and explicit RED [5,6,7].

Unlike explicit feedback, implicit feedback does not involve an explicit "send" or transmission of feedback signals. The implicit feedback (based on time-out or packet loss events) can be detected by either the sender or the receiver. For example, TCP's slow-start and congestion avoidance algorithms rely on packet loss as an implicit indication of network congestion [15]. When the source of the implicit indication is the network, the scheme is typically classified as a form of gateway congestion control.<sup>1</sup>

It is also possible to implement congestion avoidance at the endpoints based on implicit feedback such as changes in packet transit times. The most well known example of this class of congestion control is TCP/Vegas [1,2,6]. However there have been other proposals: Wang and Crowcraft's Tri-S and Dual algorithms [16,17], Haas's Adaptive Admission Congestion Control algorithm [8], and IBM's Adaptive Rate-based (ARB) protocol [12]. Of all of these algorithms, ARB is the most widely deployed as it is the congestion control scheme used in the latest release of SNA [14].

In this report, we study end-to-end congestion avoidance based on implicit feedback for TCP/IP networks. In particular, we focus on the effectiveness of

---

<sup>1</sup> TCP congestion avoidance based on a simple drop-tail router packet drop policy is usually considered to be end-to-end control. If the router participates more actively in congestion management, (e.g., RED or explicit congestion indications), then the router augments the base TCP end-to-end congestion avoidance algorithm with gateway congestion control.

three algorithms (TCP/Vegas, Dual and ARB) that use change in packet delay as an indication of the level of network congestion. The goal is to identify and explore the challenges associated with this form of congestion control. We conclude this paper with a discussion of key issues associated with end-to-end congestion avoidance and identify future work.

## 2 ANALYSIS OF END-TO-END CONGESTION AVOIDANCE

In this section, we study the congestion avoidance algorithms used by TCP/Vegas, Dual and IBM's ARB protocol. We feel that a study of these algorithms exposes the key issues. We are interested in two network environments: first, an environment where the protocol under observation competes only with similar connections; second, a best-effort IP network where the scheme under investigation must compete with any IP traffic (i.e., from other TCP or UDP connections). To help focus on congestion avoidance, we use TCP/Vegas as the base protocol and either exchange or integrate pieces of the other schemes into TCP/Vegas. This approach allows us to understand the tradeoffs of the different congestion detection schemes without clouding the discussion due to other protocol differences. Our simulation model is based on the ns simulation (v 1.4) which includes a TCP/Vegas model [4].

Figure 1 illustrates the network topology used in the simulations. Roughly half of the simulations use the simple LAN-WAN-LAN involving router's R2 and R3. In this case, the WAN is approximately T1 speed with a propagation delay of 50ms. The other simulations use the multi-hop environment provided by routers R1, R2, R3 and R4. In this case, the bottleneck link is the T1 hop. The WAN link is approximately T1 speed with a propagation delay of 50ms. All packets are 1400 bytes. We use a combination of bulk traffic source models (i.e., ftp traffic) and on/off bursty sources.

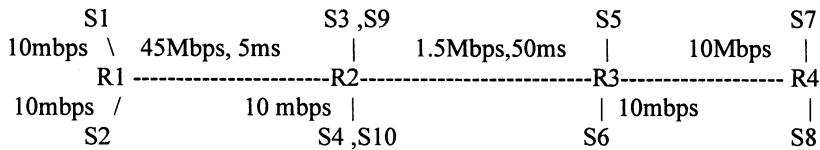


Figure 1

### *TCP-Vegas*

The aspect of TCP/Vegas that is of interest to us is the Congestion Avoidance Mechanism (CAM). CAM monitors throughput, comparing a measured throughput with an expected throughput. The *Throughput\_Diff* is the difference between an *Expected\_Throughput* and an *Actual\_Throughput*. The *Expected\_Throughput* is the current window divided by the *BaseRTT*, where the latter is the minimum round trip time observed by the connection (which

should converge to the uncongested round trip time). As long as the *Expected\_Throughput* is accurate, the  $Throughput\_Diff * BaseRTT$  (the *Diff*) is an estimate for the amount of extra data that the connection has in transit. Vegas attempts to estimate the amount of extra data and maintain the “right” amount in the network. By keeping some amount of data queued in the network, Vegas hopes to keep network utilization high.

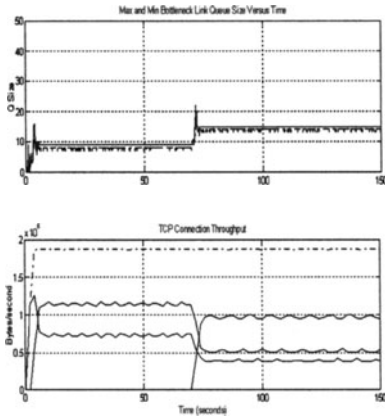


Figure 2

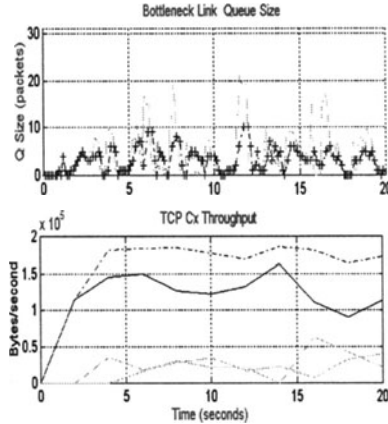


Figure 3

Figure 2 illustrates three bulk transfer Vegas connections competing over the T1 WAN hop illustrated in Figure 1. All three connections flow from R2 to R3. The top graph shows the queue level of R2’s output queue. The solid line plots the maximum queue level sampled every .1 second time interval and the dashed line plots the minimum queue level during each sample period. The router queue depth is 50 packets ensuring that packet loss does not occur.

The three solid curves in the lower graph of Figure 2 plot the throughput (measured in bytes per second each 1 second) of each connection. The dashed-dotted line plots the utilization of the bottleneck link. Vegas clearly utilizes the available bandwidth, however there is a fairness problem. The first 2 connections start at time 0 and converge to uneven shares of available bandwidth. Once the third connection starts, it obtains a significantly larger share of bandwidth than the other connections. The behavior is the same if we add a random delay in the TCP sender in the range of  $[0, 6\text{ms}]^2$ . Therefore, the problem is not a phase effect. Initially, we thought the problem might be caused by the “no change” state that Vegas tries to reach (i.e., in between the *alpha* and *beta* thresholds). We set the thresholds equal (i.e., set both *alpha*

<sup>2</sup> We modified the ns TCP/Vegas implementation such that timestamps are recorded before the send delay. This injects “noise” in the Vegas congestion assessment. The TCP sender is coded such that if it has multiple packets to send (i.e., if an Ack causes the *cwnd* to increase), the random send delay occurs once prior to the burst. After the delay expires, the sender is allowed to send all packets in the burst instantaneously.

and *beta* to a value of 3) to eliminate occurrences of a “no change” CAM decision. This led to similar results as in Figure 2.

There are actually two problems that explain the behavior observed in Figure 2. For static networks (i.e., a network that consists of constant Vegas senders that reach some steady state), it is likely that the system will converge to an unfair state. Once in this state, the increase/decrease algorithm is not sufficient to move the system to a fair state. The second problem is that once the system enters steady state, the existing connections have pushed the network such that there is some amount of sustained queueing. Late starting connections can not detect the congestion and instead will add to the existing congestion forcing the existing connections to reduce their send rates. Therefore, in a congested system that has reached steady state, late starting connections will obtain an unfairly large share of available bandwidth.

A more dynamic network environment will reduce the probability of the system entering a steady state. Figure 3 illustrates the same scenario as in Figure 2 except that connections 2 and 3 are configured to use an on/off bursty traffic source rather than an ftp source. The bursty sources have mean rates on the order of 100000 bps. The lower curve shows that the two bursty connections (the two light lines) obtain bandwidth from connection 1 (the dark line) in a fair manner. The dashed-dotted line illustrates that Vegas is able to utilize on the order of 90% of the available bandwidth. The upper plot of Figure 3 shows the minimum queue level at the bottleneck link (the light dashed line). The “+” marks represent the Vegas sampled queue level (i.e., the *Diff*).

In the lightly loaded network illustrated by Figure 3, CAM responds to congestion well although it does not track network congestion precisely. As described earlier, Vegas detects congestion by looking for changes in throughputs. The *Actual\_Throughput* is based on the number of packets transmitted during the past RTT and on the actual RTT sample. The original Vegas proposal suggested that one packet each RTT be selected to probe for congestion. The Vegas implementation based on NetBSD actually uses the average RTT's for all packets (since Vegas times all packets) that were acknowledged during a measurement interval [1]. The idea being to filter noise associated with individual RTT samples.

In the best case, an end-to-end delay measurement algorithm tracks queueing at the bottleneck link caused by the aggregate traffic from all sources. In the worst case, the algorithm tracks queueing caused only by the connection itself. The following analysis shows that Vegas does not accurately track neither network level queueing nor queueing caused by the Vegas sender. In certain situations, the algorithm's congestion assessment effectively becomes meaningless as the *Diff* value converges to a fixed window value. The latter case explains previous analysis results of Vegas that conclude that during periods of heavy congestion, the scheme digresses to TCP/Reno behavior[2].

First, we show a simple example (a single Vegas connection with no competition) where CAM accurately tracks the queue level at the bottleneck

link. Using the network shown in Figure 2 as an example, assume that a single Vegas always has 1400 byte packets available to send. At the point where the T1 link is fully utilized, the Vegas *cwnd* is 13.4 packets (i.e., a bandwidth-delay product of 13.4 packets). The *expected* throughput is naturally 1.5Mbps and the *actual* throughput should be the same. The next RTT, the *cwnd* will be 14.4. The *expected* throughput will be just over 1612800 bps. Assuming exactly 1 packet experienced a waiting time of a packet transmission time at the T1 link, the *rtt* should be  $.1 + 1400 \cdot 8 / 1.5\text{Mbps}$  or .1075 seconds. The *Actual\_throughput* will therefore be 14.4 packets / .1075 seconds or roughly 1.5 Mbps. Multiplying the difference in throughput by the *BaseRTT* corresponds to a *Diff* in packets of roughly 1400 bytes or 1 packet.

When Vegas competes with a low to moderate amount of traffic, the following helps explain the behavior of the *Diff* samples as illustrated by the upper curve in Figure 3. The *Diff* value in bytes can be written as:

$$\text{Diff} = (W/\text{BaseRtt} - W'/\text{Rtt}) * \text{BaseRtt}$$

where  $W$  is the current window,  $W'$  is the amount of data sent during the measurement period, and  $Rtt$  is the current RTT sample. As long as the sender has data to send, no packets are lost and the receiver ACKs each packet,  $W'$  will be the current window,  $W$ . Therefore:

$$\begin{aligned} \text{Diff} &= W - W * \text{BaseRtt} / \text{Rtt} \\ \text{Diff} &= W(1 - \text{BaseRtt} / \text{Rtt}) \end{aligned}$$

Clearly, *Diff* is 0 when  $\text{BaseRtt} = \text{Rtt}$ , and is positive when  $\text{Rtt} > \text{BaseRtt}$ . Also note that the  $\text{Rtt} = \text{BaseRtt} + Q_t$  where  $Q_t$  reflects queueing delays. The upper bound of  $(1 - \text{BaseRtt}/\text{Rtt})$  is 1 which means that the largest *Diff* value that can ever be observed is the current window size. The  $(1 - \text{BaseRtt}/\text{Rtt})$  term essentially grows linearly with increasing  $\text{Rtt}$ , however the  $W$  will also decrease in response to a positive *Diff*. As the  $\text{Rtt}$  increases, the rate of increase of the *Diff* is dampened as  $W$  decreases. This explains the behavior of the *Diff* curve in Figure 4. As the queue builds, the term  $(1 - \text{BaseRtt}/\text{Rtt})$  increases however the *Diff* value might actually decrease as  $W$  decreases in response to the congestion. The scheme is fair in the sense that a connection with a high bandwidth will react more aggressively to increases in  $\text{Rtt}$ 's. However, as the following discussion will show, the scheme loses its effectiveness when operating in periods of heavy congestion.

Vegas does not decrease the window if  $\text{Diff} \leq \text{beta}$ . The point where the algorithm stops decreasing the window is:

$W_{\min} (1 - \text{BaseRtt}/\text{Rtt}) = \text{beta}$  where  $W_{\min}$  is the lowest window value that is to be used.

$$W_{\min} = \text{beta} / (1 - \text{BaseRtt}/\text{Rtt})$$

For large  $\text{Rtt}$ 's with respect to  $\text{BaseRtt}$ , the  $W_{\min}$  approaches  $\text{beta}$ . We will see that in heavily congested networks (i.e., networks where there is a large amount of sustained queueing), the Vegas algorithm effectively holds the sender to a fairly constant window of  $\text{beta}$  packets (3 for our simulations). We

refer to this as the overload state. Given this, the throughput of Vegas (in packets per second) in heavily congested networks can actually be predicted as follows:

$$Vegas\_throughput = \beta / (BaseRtt + (Q_{avg} * MSS * 8) / Cbl)$$

where  $Q_{avg}$  is the average queue length sampled each RTT, the  $MSS$  is the maximum segment size and the  $Cbl$  is the bottleneck link capacity.

Figure 4 illustrates the results of a simulation run based on the multi-hop network shown in Figure 1 with 3 ftp TCP/Vegas connections (S1, S2 and S3 all to R4) compete with 3 on/off bursty TCP/Vegas connections (S4, S9 and S10 to R3). The bottleneck router can buffer up to 50 packets. The simulation is intended to capture the behavior of a heavily congested network. The minimum and maximum queue levels of the T1 link shown in the upper plot (the dark and the dotted line respectively) demonstrates that CAM is not able to prevent sustained congestion from occurring. The “+” marks plot the *Diff* samples from the first Vegas connection. The lower plot shows the three ftp Vegas connection’s throughputs (we do not show the throughputs of the bursty connections, only the effective utilization). The throughput curves illustrate the bias towards late starting connections (connection 1 gets a lower share of available bandwidth once the system converges). If we start a fourth ftp connection at time 20 seconds, it would obtain a much larger share of bandwidth than any of the other connections.

The connections with accurate *BaseRTT* values (i.e., the two connections that start at time 0 and 2 respectively in Figure 4) converge to a window of about 4 packets with a sustained throughput close to the predicted throughput of a Vegas connection that has reached the overload state. Late starting connections might not reach the overload state since they will be much more tolerant of congestion (since the *BaseRTT* will include the sustained congestion levels). If too many Vegas connections are in the overload state at the same time, packet loss will occur and the behavior of each connection will digress to TCP/Reno (i.e., oscillating window values). Figure 5 shows a more extreme environment than that depicted in Figure 4. The router buffer capacity is reduced to 20 packets and we add two additional ftp Vegas connections (that start at time 0 and 2 seconds respectively). The packet loss rate is quite high, about 6.2%. Each Vegas connection experiences time-outs (in the range of 2 to 15) contributing to *cwnd* oscillations. This demonstrates the tendency for Vegas to digress to Reno behavior in high packet loss environments.

In the analysis presented so far, we have observed Vegas in an environment where it competes only with other Vegas connections. It is also interesting to see how Vegas behaves when competing against other TCP/Reno connections. Figure 6 illustrates a simulation with one Vegas connection (the first connection that starts at time 0) and two TCP/Reno connections that start at time 5 and 70 seconds respectively. All connections are configured with ftp traffic sources. The simulated network involves the T1 hop between R2 and R3 as illustrated in Figure 1. The lower curve shows synchronized behavior similar to that seen in Figure 2. The Vegas connection increases its throughput until time 2 seconds when the second connection starts. The

second connection (i.e., the Reno connection) reaches its maximum window (36 packets) causing additional sustained queueing that forces the Vegas connection to a state of low throughput. The system remains locked in this state until the second TCP/Reno connection forces packet loss after time 70. Note that after time 70 the system reaches a new synchronized state. However, since the queue levels stay consistently high, the Vegas connection is never able to obtain its fair share.

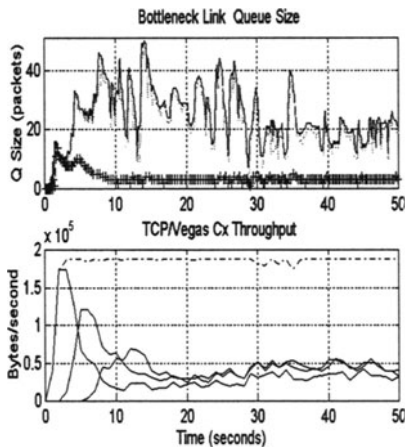


Figure 4

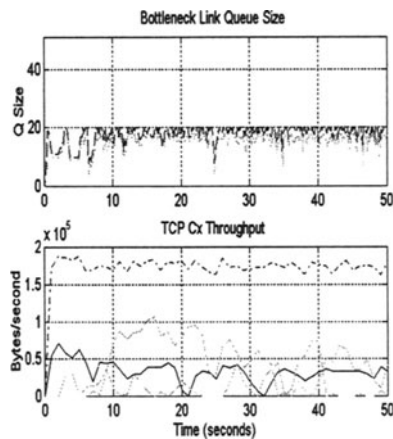


Figure 5

Figure 7 illustrates a more realistic simulation than the previous. Based on the network shown in Figure 1, a Vegas ftp connection from S1 to R4 starts at time 0. Two on/off bursty Reno sources flow from S3 and S4 to R3 that produce bursty cross traffic. The solid dark line in the lower curve illustrates the Vegas throughput and the lighter lines represent the throughput of the two on/off connections. The behavior is similar to the earlier Vegas case where we concluded that CAM is effective at tracking and controlling congestion in a mildly congested network. If we do the same simulation except use all Reno connections, the performance of the ftp connection is similar to the Vegas run. Both achieve roughly the same throughput, neither experience time-outs. One significant difference, however is that the queue levels are more controlled in the Vegas case than in the all Reno simulation.

Figure 8 illustrates how an ftp Vegas connection (the dark link in lower curve) competes with two ftp Reno connections and 3 bursty Reno. The behavior is as observed in the USC analysis where Reno steals bandwidth from Vegas in head-to-head competition simply because Vegas is more sensitive to congestion than Reno[1]. Confirming our earlier analysis, the Vegas connection is essentially limited to the  $W_{min}$  of 3 which, assuming an average queue level of 35 packets (by inspection from Figure 8), should lead to a throughput of 93,000 bps. Based on the Vegas throughput curve we see a Vegas throughput of roughly 120,000 bps.

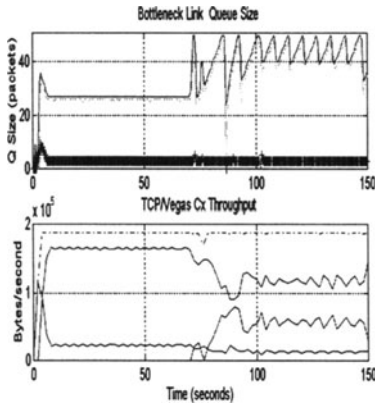


Figure 6

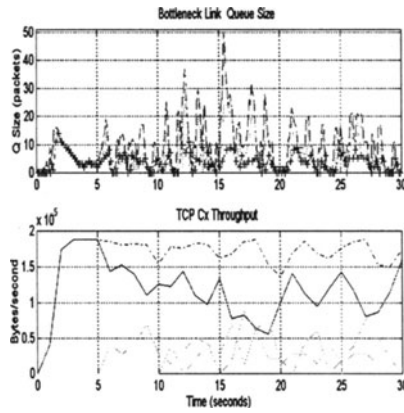


Figure 7

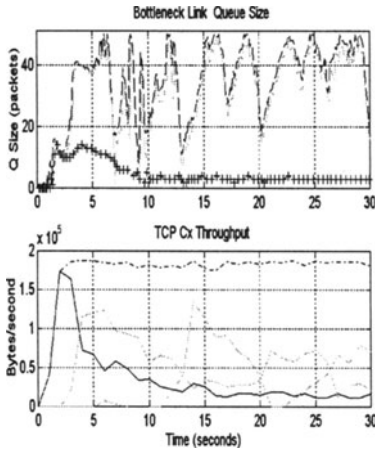


Figure 8

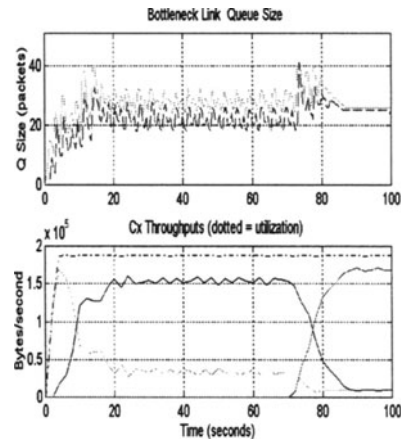


Figure 9

### Dual

Dual offers a congestion detection scheme that monitors changes in measured round trip times. It tracks the minimum observed RTT ( $RTT_{min}$ ) and the maximum observed RTT ( $RTT_{max}$ ) over the lifetime of the connection. In most network environments, over time these variables converge to the static round trip delay due to propagation delays and to the variable delay representing queuing delay respectively. Each time TCP performs its round trip time calculation, the sampled RTT is compared with a delay threshold defined as:

$$D_i = (1-\alpha)D_{min} + \alpha * D_{max}.$$



Dual was designed to extend TCP's congestion avoidance and slow start algorithms with the goal of reducing the oscillations caused by slow start and congestion avoidance. The change to TCP to implement Dual is trivial. Slow start and congestion avoidance remain unchanged except every other round trip time, the following comparison is made:

*If (rtt > Di)*  
     *Cwnd -= min(cwnd, wnd)/8;*

We have simulated a modified version of the Dual algorithm essentially replacing Vegas's CAM with Dual's congestion detection scheme. More specifically we modified Vegas as follows to implement Dual:

- Continue to time each packet and to aggressively retransmit packets. However, the algorithm will not track changes in throughput and react to decreases.
- The increase algorithm of TCP/Dual is essentially the same as that used by Vegas. We continue to do exponential growth only every other round trip time during slow start. However the decrease algorithms differ. While Vegas decreases the window linearly if the congestion threshold is exceeded, TCP/Dual decreases the window by 12.5% if a round trip time exceeds the *Di* threshold.
- The Dual algorithm indicates that when a timeout occurs, we should reset the *Dmin* and *Dmax* values. While this might be useful to handle path switch situations, we chose not to implement this.

Figure 9 shows the behavior when three Dual connections compete (using the T1 network from Figure 1). Due to sustained congestion, late starting connections will have an incorrect threshold causing the connection to act more aggressively than connections with accurate threshold values. By the time the third connection starts, the connection is not able to differentiate between propagation delay and queueing delay. If we do the same simulation, however reduce the router buffer size to 10 packets, the throughput of the three connections converges quickly, although several packets are dropped as the system converges. The difference is that Dual's 12.5% rate reduction is sufficient to clear a small amount of queueing (less than 10 packets). In the case pictured in Figure 9, a 12.5% rate reduction is not sufficient to clear the queue. The algorithm requires the queue levels to on average remain close to 0 so that late starting connections can obtain an accurate *Dmin* value.

Figure 10 shows three Dual connections (1 ftp, 2 bursty) using the multi-hop network between routers R1 and R4 in Figure 1. The top curve shows only the minimum queue level sampled every .02 seconds. The results show that Dual utilizes the bandwidth but has sustained congestion. To test the sensitivity of the algorithm to noisy RTT samples, we configured a random send delay in the range [0-2ms]. There was no difference in behavior. The reason is straightforward: Dual's threshold is on the order of ½ the buffer range. At 16Mbps, a 2 ms delay in the RTT sample corresponds to roughly 3 packets. It would take a much larger delay variation to be detected by Dual. In fact, it is

not until we increase the random send delay range to [0-.01seconds] before performance deteriorates as Dual reacts prematurely to the noisy samples.

Figure 11 shows 1 Dual connection (the dark line in the lower curve) competing with 2 Reno connections (the two light lines that start at time 5 and 10 respectively). Clearly the Reno connections are more aggressive. The system falls into a synchronized state such that the Reno connections utilize the majority of bandwidth. The problem is that the Dual threshold needs to be adjusted (i.e., increased) to compete fairly with Reno.

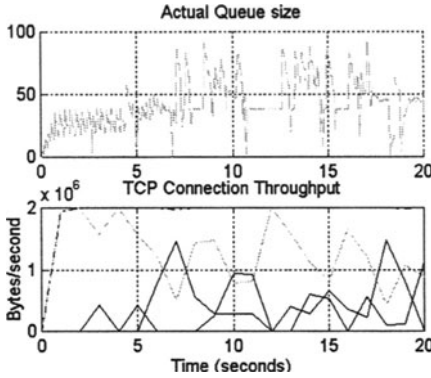


Figure 10

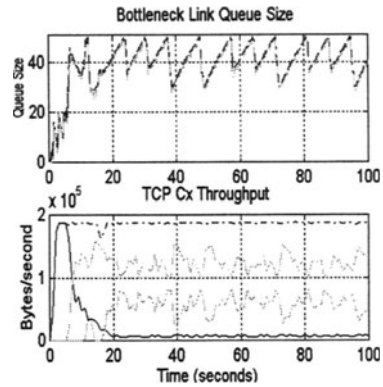


Figure 11

## ARB

The Adaptive Rate-based protocol is an end-to-end congestion avoidance scheme used in IBM's Rapid Transport Protocol (RTP) [14]. ARB is a closed-loop, preventive, rate-based congestion control scheme. ARB employs a distributed algorithm that is implemented at the endpoints of an RTP connection. Each endpoint consists of an ARB sender and an ARB receiver. The ARB sender periodically queries the receiver by sending a rate request to the ARB receiver who responds with a rate reply message. The time between successive transmissions of rate request packets is defined as a *measurement\_interval*. The *measurement\_interval* is typically on the order of a round trip time although to minimize processing overhead, RTP products typically have a minimum *measurement\_interval* on the order of .1 second.

The ARB receiver monitors changes in the delay experienced by sequential rate request packets. It maintains its own version of the time between successive rate request packet arrivals (the *receivers\_measurement\_interval*) and compares to the sender's *measurement\_interval* (which are contained in the rate request packets). A positive difference corresponds to additional delay experienced by the probe packet. Likewise, a negative delay corresponds to less waiting time experienced by the probe as compared with the previous probe packet. ARB assumes that trends in the delay change values (i.e., a

running total of delay change samples) are reflective of the current level of congestion in the one-way path between the sender and the receiver. For example, if the *total\_delay* is 0, then packets that arrive at the receiver have not experienced congestion. If the *total\_delay* is 50 mseconds, then each packet would have experienced 50 mseconds of queueing delay.

The receiver translates the *total\_delay* into an estimate of the amount of queueing at the bottleneck link. In an SNA network, the receiver learns the slowest link speed in the path via the RTP connection setup protocol (known as the *max\_bandwidth*). The *queueing\_estimate* is simply the *total\_delay* divided by the *max\_bandwidth* (this gives a total queueing in bytes, RTP converts this into a number of 1000 byte packets). The fundamental control decision behind ARB is made by the receiver based on a threshold of allowed queueing. The sender is allowed to increase its rate as long as the *queueing\_estimate* is less than 1 packet. If the *queueing\_estimate* is between 1 and 10 packets, the sender is instructed to restrain (i.e., not to change its sending rate). If the *queueing\_estimate* is between 10 and 40 packets, the sender is instructed to reduce its send rate. Finally, if the *queueing\_estimate* exceeds 40 packets, ARB assumes that this is noise and tells the sender to restrain. The sender adjusts its send rate based on information received in the rate reply message. Refer to [12] for a detailed description of the ARB algorithm.

In the remainder of this section, we present and study a congestion avoidance scheme that extends TCP with the essence of ARB. We are most interested in the ARB's congestion detection scheme and in the receiver's control decision logic. We leave the study of rate control for future work. The key design points of TCP/Arb include the following:

- Keep all aspects of TCP/Reno and TCP/Vegas except remove CAM (i.e., we still want Vegas to time all packets and to remain more aggressive than Reno with retransmission). Therefore slow start and congestion avoidance are preserved.
- The sender periodically forwards a probe packet (using TCP options) that contains a measurement request signal along with the *senders\_measurement\_interval* (which is simply the amount of time since the sender last issues a measurement request packet).
- The receiver responds to measurement request packets from the sender by calculating the *receivers\_measurement\_interval* and obtains a *delay* value (by subtracting the receivers and senders measurement intervals). The receiver maintains the commulative delay in a variable called *total\_delay*. The receiver also monitors the observed throughput ( $\text{throughput} = \text{byte\_count} / \text{receivers\_measurement\_interval}$ ) and maintains the highest throughput observed (*max\_bandwidth*). The receiver estimates the level of queueing ( $\text{queueing\_estimate} = \text{total\_delay} / \text{max\_bandwidth}$ ). The receiver inserts a *rate\_command* message in the Ack (again using TCP options). The command (RATEINCREASE, RATEDECREASE or RESTRAIN) is based on the receiver's *queueing\_estimate* with respect to the ARB thresholds. Therefore, if the

receiver's *queueing\_estimate* is less than or equal to 1 packet, the receiver issues a RATEINCREASE. If the *queueing\_estimate* is between 1 and 10 packets, the receiver issues a RESTRAIN. An RTP receiver will instruct the sender to RATEDECREASE when the *queueing\_delay* exceeds 10 packets.

- When the sender receives a *rate\_command* of RATEINCREASE, the sender increases as normal (i.e., by incrementing the *cwnd* as required during either slow start or congestion avoidance). The exception is during slow start, the sender increases the send rate exponentially every other round trip time (as done in Vegas). If the sender receives a RATEDECREASE command, the sender reduces its *cwnd* by 50%. If the sender is in slow start when it receives a RATEDECREASE command, it moves to congestion avoidance by setting the *ssthresh* to 2 (TCP/Vegas also does this).
- Probe packets are not issued during periods of recovery. Therefore, during periods of heavy packet loss, the scheme digresses to base TCP.

Figure 12 illustrates 2 TCP/Arb connections competing for bandwidth using the T1 hop between routers R2 and R3 illustrated in Figure 1. The upper curve plots the maximum queue level (dark dashed line), the minimum queue level (light dashed line) and the second connection's *queueing\_estimate* (the "+" marks). Notice the the connection underestimates the queueing level. The lower curve of the figure illustrates that the second connection (the light line that starts at 2 seconds) obtains the majority of the bandwidth. Because the second connection underestimates queueing, it obtains a larger share of available bandwidth.

There are two factors that contribute to connection two's *queueing\_estimate* error. First, the connection begins during a period of sustained congestion. The minimum sustained congestion experienced by the second connection is on the order of 3 packets. The connection can not detect the queue buildup which contributes the majority of the *queueing\_estimate* error.

The second factor that contributes to the error is due to an inaccurate *max\_bandwidth* estimate. The original ARB converts the *total\_delay* to an estimate of the amount of data queued in the path based on the *max\_bandwidth*. Ideally, the *max\_bandwidth* is reflective of the bottleneck link speed. The TCP/Arb receiver monitors observed throughput roughly each round trip time in an attempt to estimate the *max\_bandwidth*. Several factors combine to make the connection tend to underestimate the *max\_bandwidth*. Clearly, the *max\_bandwidth* is affected by the connection's actual sending rate. If the sender can not fill the one-way pipe (e.g., if the maximum window size is too low or if the source does not have enough data to keep the sender busy) the *max\_bandwidth* will be low. Additionally, if the sender is paced by the congestion control then the *max\_bandwidth* will tend to measure the connection's available bandwidth. However, due to the "ack clumping" phenomenon, TCP is actually quite bursty. For example, assume that a TCP connection has achieved a sustained throughput of  $\frac{1}{2}$  the bottleneck link speed.

The connection can actually be modeled as an on/off source, bursting at some high rate (determined by the ack return rate) for a certain duration (depending on the *cwnd*) and then “off” as the bottleneck link is used by other connections [13]. Once the burst is large enough to fill the one-way pipe, the connection will be able to get an accurate *max\_bandwidth* estimate. Due to slow start’s blind exponential growth, the typical connection (as long as it is not window constrained) will be able to get a fairly accurate estimate of the *max\_bandwidth*.<sup>3</sup>

Figure 13 illustrates the *max\_bandwidth* for the two connections shown in figure 12. The one-way pipe size is about 7 packets and if filled by the first connection in just over 1 second. At about time 3.3 seconds, Figure 13 indicates the queue level increases by 7 packets for a total congestion level of roughly 10 packets. However, connection 2 only detects an additional 2 packets of queueing. It actually observes the correct increase in *total\_delay*. However, as can be seen in Figure 14, the *max\_bandwidth* at this time is still low which leads to more error (in addition to the error caused by the sustained congestion). Once in steady state (after time 6 seconds), there are roughly 6-7 packets of sustained queueing. The second connection’s *queueing\_estimate* observes 2-3. Most of the error is caused by the sustained queueing, however an additional 1 packet error results from the second connection’s *max\_bandwidth* being off by roughly 25%.

Figure 14 shows 1 ftp TCP/Arb connection (the dark line) competing with 2 bursty TCP/Arb connections over the single hop T1 network shown in Figure 1. Comparing Figure 14 to the equivalent Vegas simulation run (Figure 3), the throughput of the TCP/Arb and Vegas ftp connections are virtually identical. The difference lies in the queue levels as TCP/Arb controls the queue levels more effectively than Vegas.

Figures 15 illustrates how TCP/Arb behaves when competing with TCP/Reno connections. Figure 15 shows 1 ftp TCP-Arb connection (the dark line) compete with 2 ftp TCP/Reno connections. Given that TCP pushes the network well beyond ARB’s threshold level, competing TCP connections generally “beat down” TCP/Arb connections. Because TCP/Arb accurately tracks queueing, it obtains only a fraction of available bandwidth. It is possible to tune the ARB region thresholds such that TCP/Arb connections compete with Reno connections. The problem is that the proper thresholds depend primarily on the queueing behavior at the bottleneck link which clearly can not be statically predicted.

---

<sup>3</sup> A further improvement is to use packet pair and have the receiver monitor the difference between arrival times of successive packets [10].

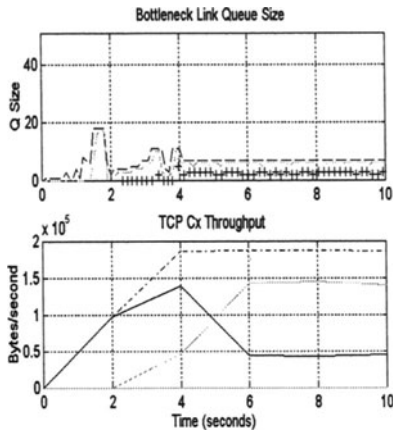


Figure 12

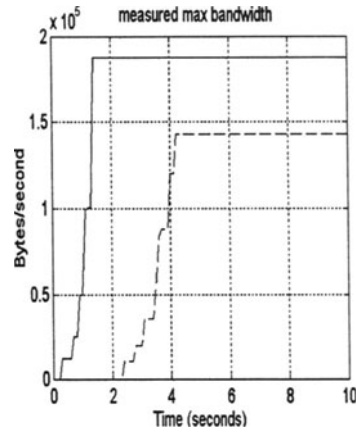


Figure 13

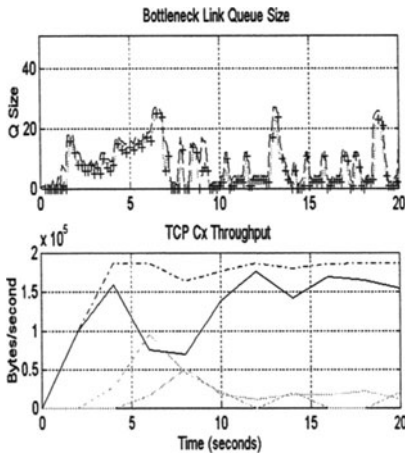


Figure 14

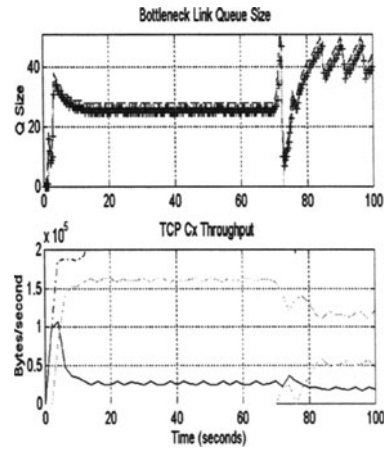


Figure 15

### 3 CONCLUSIONS

Based on the discussion and analysis presented in this report, we conclude with the following observations

- Congestion avoidance works properly only if the network does not suffer from sustained congestion. It is unclear how prevalent sustained congestion actually is in the Internet. However, we would like to believe that such conditions are the exception rather than the norm
- The scheme should track bottleneck queue behavior as accurately as possible. Changes in delay are the purest indicator of queue behavior. Converting a commulative delay into an estimate of queueing at the bottleneck link adds error into the control decisions as it must be based on an estimate of the bottleneck link speed.
- Path switches in the Internet are fairly common. Clearly, an algorithm such as TCP/Vegas that records the minimum round trip time measurement as a baseline can lead to disastrous results. One solution might be to deduce a path switch (via a burst of packet loss) and to reset the baseline round trip time measurement. However, clearly, path switches are problematic for end-to-end control schemes.
- Although not we did not show the simulations, we find that a one-way congestion estimate is superior to a scheme that measures congestion in both directions. For example if the return path in a TCP/Vegas connection has significantly more delay (due to congestion, different propagation delay or different link speeds), it will not fully utilize available bandwidth in the forward direction. A scheme like TCP/Arb that measures delay in a one-way direction eliminates the error.
- We have not shown the corresponding analysis, however we find that noise in the delay samples can cause sometimes significant error in the control decisions. Therefore the congestion samples must be filtered in some manner, or more likely, the threshold (i.e., the point at which the controller decides to react to congestion) must be adaptive to support changing conditions. Crucial issues involved with finding the optimal threshold point include:
  - The scheme must learn the noise floor. The goal is to find the minimum threshold such that the queue levels are contained. Clearly this conflicts with the goal of high link utilizations.
  - When competing against TCP (or other greedy protocols), a congestion avoidance scheme needs to learn the upper bound for the threshold.
  - More frequent feedback indications can be used to converge more quickly to the optimal threshold level.

In this paper we have shown that in certain environments and conditions, an end-to-end congestion avoidance algorithm based on packet transit delay measurements can be beneficial by avoiding packet loss and stabilizing response times. We have also identified several factors that can cause these algorithms to not work correctly. Unfortunately, today's best effort Internet can not guarantee that these factors such as path switches or sustained congestion will not occur.

## 4 REFERENCES

1. J. Ahn, P. Danzig, Z. Liu, L. Yan, "Evaluation of TCP Vegas: Emulation and Experiment", ACM SIGCOMM95.
2. O. Ait-Hellal, E. Allman, "Analysis of TCP-Vegas and TCP-Reno", ICC, June 1997.
3. L. Brakmo, S. O'Malley, L. Peterson, "TCP Vegas: New Techniques for Congestion Detection and Avoidance", ACM SIGCOMM94, 1994.
4. K. Fall, S. Floyd, S. McCanne, network simulation ns, "<http://www-mash.cs.berkeley.edu/ns/ns.html>", 1996.
5. S. Floyd, "TCP and Explicit Congestion Notification", ACM Computer Communications Review, October 1994.
6. S. Floyd, K. Ramakrishnan, "A Proposal to add Explicit Congestion Notification (ECN) to Ipv6 and to TCP", Internet Draft, Nov 1997, <draft-kksjf-ecn-00.txt>.
7. S. Floyd, V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, August 1993.
8. Z. Haas, "Adaptive Admission Control", ACM SIGCOMM91, 1991.
9. V. Jacobson, "Congestion Avoidance and Control", ACM SIGCOMM88, 1988.
10. S. Keshav, "Packet-Pair Flow Control", <http://www.cs.cornell.edu/skeshav/doc/94/2-17.ps>.
11. A. Mankin, K. Ramakrishnan, "Gateway Congestion Control Survey", RFC 1254, 1991.
12. J. Martin, A. Nilsson, "Congestion Control in HPR", IEEE GLOBECOM97.
13. J. Martin, et. Al., "A Comparison of TCP/Reno and RTP Transport Protocols", IBM Technical Report TR-29.2337.
14. J. Nilausen, "APPN Networks", Wiley 1994.
15. R. Stevens, TCP/IP Illustrated, Volume 1, Addison-Wesley, 1994.
16. Z. Wang, J. Crowcroft, "A New Congestion Control Scheme: Slow-start and Search (Tri-S)", ACM Computer Communication Review, V21 #1, January 1991.
17. Z. Wang, J. Crowcroft, "Eliminating Periodic Packet Losses in the 4.3-Tahoe BSD TCP Congestion Control Algorithm", ACM Computer Communication Review, April 1992.

## 5 BIOGRAPHY

Jim Martin has been a software developer for IBM for the last 7 years. He is currently a PhD student at North Carolina State University. His research has focused on congestion control. Specifically, he is researching transport protocols with respect to congestion control for the Internet.



# **Part Ten**

---

## **Flow and Congestion Control**

# A Rate Based Back-pressure Flow Control for the Internet

*Carlos M. Pazos and Mario Gerla*  
*Computer Science Department*  
*University of California, Los Angeles*  
*405 Hilgard Ave., Los Angeles, CA 90024*  
*{pazos,gerla}@cs.ucla.edu*

## Abstract

The Internet has traditionally relied on end-to-end congestion control performed at the transport layer by TCP. In this paper, we discuss the limitations of this approach to address the large number of flows and the large delay-bandwidth product scenarios typical of next generation Internets. We propose a link layer back-pressure flow control which can be applied to Internet backbones over ATM. More precisely, we use the ABR service and flow control and we turn routers into virtual sources (VSs) and virtual destinations (VDs) for the ABR control loop. We introduce a VS/VD “behavior” that implements a rate based back-pressure flow control and that addresses max-min fairness.

## Keywords

Internet Backbones, ABR Service, Back-pressure Flow Control.

## 1 INTRODUCTION

Following the rapid evolution of Internet services in recent years, the Best Effort (BE) service is no longer adequate to address the requirements of new services, some of which have real time (RT) constraints and thus require resource commitment from the network in order to implement Quality of Service (QoS) guarantees. The IEFT Internet Integrated Services (ISS) working group is investigating new services and features to allow the Internet to transport multimedia traffic (Braden *et al.* 1994).

As a result, we are bound to experience an ever increasing demand for transmission resources, already a reality given the exponential growth of current Internet traffic. The conventional approach to address this problem has consisted on adding faster links to Internet backbones, upgrading the available pool of resources. While this course of action satisfies the Internet craving for bandwidth, it also makes it more difficult to control traffic and prevent congestion. With faster transmission rates, we are unavoidably faced with

problems due to large delay-bandwidth products on backbone links (a huge number of packets is in transit on these links).

The RT traffic can reserve resources and it does receive priority service on routers in the backbones. For this traffic, appropriate resource allocation and efficient admission control are probably enough to avoid the effects of congestion. On the other hand, the bulk of BE traffic is transported over the backbones under TCP window flow control applied to individual sessions on the edges of the network. Namely, TCP sources adjust their transmission rates in reaction to congestion in the network, which is detected when packet are lost. Hence, actions to remedy congestion are only taken when congestion sets in. The associated reaction time is in the order of end-to-end round trip times (RTTs).

Furthermore, it has been argued (Morris 1997) that TCP alone cannot fairly and effectively control a very large number of sessions, an increasingly frequent scenario in the Internet. Increasing the bandwidth available on backbone links and/or increasing buffering space on routers can help accommodate the traffic of a very large number of sessions (Morris 1997), but TCP sessions are not generally subjected to admission control.

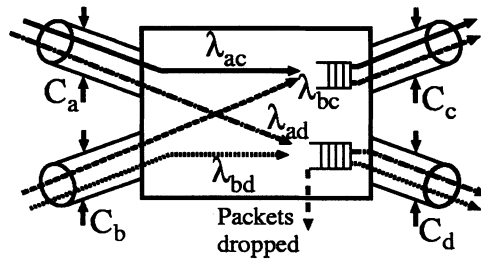
In this paper we describe a rate based back-pressure flow control that acts on the **aggregate** traffic between each pair of routers. Each router continuously notifies its neighbors of the rate it can accept from them and the back-pressure control propagates all the way to ingress edge routers. Hence, the response time is in the order of the RTT between the edge router and the point of congestion.

Our flow control approach builds on the work in (Pazos *et al.* 1997), where the use of the ATM ABR service was suggested for Internet backbones. Namely, the links interconnecting IP routers are ABR VCs, rather than the more conventional CBR or UBR VCs. With ABR, the backbone VCs are not restricted to the CBR peak rate allocation and statistical multiplexing is improved. As compared with UBR, the ABR service offers a much better protection guarantee. In this paper, we take a step further and we implement a rate based back-pressure flow control scheme. One considerable advantage of this approach is that it is almost entirely implemented at the ATM layer on routers and hence it requires no modifications to either TCP or IP protocols.

The balance of the paper is organized as follows. In section 2 we describe the congestion problem we address in this paper and in section 3 we present the network scenario we consider for the remainder of the paper. In section 4 we review current approaches for congestion control in Internet backbones and we introduce our back-pressure flow control scheme. In section 5 we present simulation results. Finally, in section 6, we make some concluding remarks.

## 2 CONGESTION IN THE BACKBONE

Consider the model shown in Figure 1. Congestion occurs when the incoming traffic ( $\lambda_{ad} + \lambda_{bd}$ ) feeding an outgoing Virtual Circuit  $VC_d$  exceeds its capacity  $C_d$ . As a result, packets may be dropped, impacting effective network utilization and compromising performance. In order to avoid such losses, we need to selectively slow down the traffic flows  $\lambda_{ad}$  and  $\lambda_{bd}$  at the respective sources. This is clearly a hard problem to solve unless an end-to-end rate based (or credit based) feedback mechanism such as the ABR flow control (ATM Forum 1996) is employed. However, since most WAN TCP sessions transfer only a few kilo-bytes of data (Paxson 1993), and since convergence to the fair share of the available bandwidth may take a number of RTTs (Jain *et al.* 1996), the efficiency of the ABR flow control may be compromised. The Explicit Congestion Notification (ECN) approach proposed for IPv6 in (Floyd 1995) attempts to provide feedback control from the network layer (IP) to TCP. The approach in (Roche *et al.* 1995) allows an edged router to provide congestion feedback to other sending edge routers.



**Figure 1** A congested router.

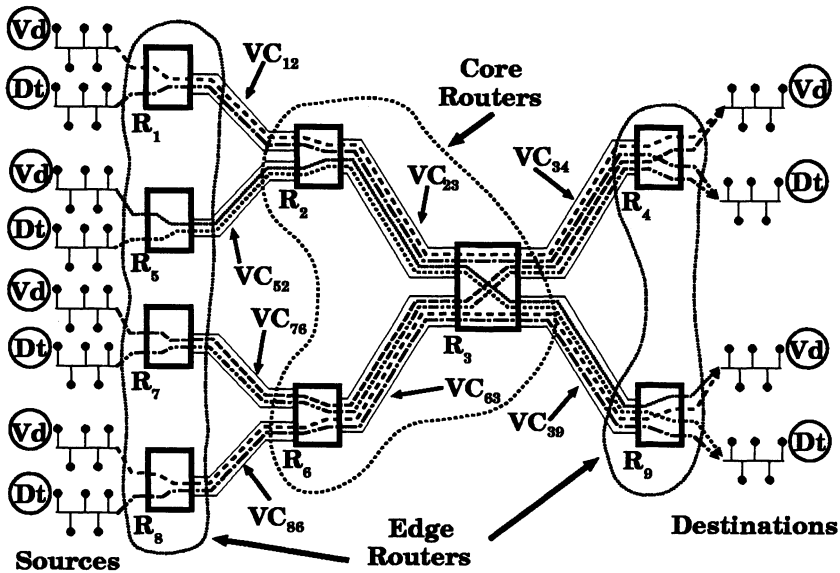
The current approach for congestion control in backbones relies on the TCP window flow control. With TCP, only those sessions suffering packet loss (experiencing congestion) reduce their transmission rate (window). In Figure 1, for instance, the sessions using  $VC_c$  are unaffected by the congestion on  $VC_d$  and the TCP flow control does not act on them. However, if the flow  $\lambda_{ad}$  is responsible for the abuse of  $VC_d$ , the flow  $\lambda_{bd}$  is also penalized in the process and the TCP mechanism cannot selectively slow down the  $\lambda_{ad}$  flow. Namely, TCP alone cannot ensure fairness (Morris 1997). A number of other unfair TCP behaviors have been addressed in (Floyd *et al.* 1991, Floyd 1991).

Hence, the best to alleviate the congestion on the router of Figure 1 and to avoid packet losses is to selectively back-pressure the flow  $\lambda_{ad}$ . Such preventive action can effectively address the large delay-bandwidth product case if we make back-pressure propagate from the congestion point all the way to the sources. Even if this is very difficult to implement, since routers do not have

end-to-end information, back-pressuring aggregate input port traffic is enough to alleviate congestion in many instances as discussed in this paper.

### 3 THE NETWORK SCENARIO

We study the congestion control problem in the IP over ATM backbone scenario. Nonetheless, we are not actually assuming ATM connectivity end-to-end, rather, we assume that Internet users reside on legacy LANs. Edge routers, with ATM connectivity, function as ingress routers for the traffic crossing from the LAN to the backbone (and vice versa), see Figure 2. Hence, in our model, the edge routers are the effective sources and sinks of Internet traffic.



**Figure 2** The network scenario.

As Figure 2 also illustrates, we consider a scenario in which the backbone transports Data (Dt) and video (Vd) traffic. In our simulation studies, we consider the Dt traffic being file transfers over TCP and the Vd traffic being compressed video sent over UDP. We place Vd and Dt sources and sinks in different legacy LANs for simplicity since we are not concerned with the individual traffic flows. Our interest is on the aggregate streams for each class on which we exercise back-pressure.

The Vd traffic is given priority over the Dt traffic on edge and core routers

by a Class Based Queueing (CBQ) (Floyd *et al.* 1995) scheduler. We also assume that RSVP and some form of flow admission control are employed to allocate the Vd traffic flows appropriate bandwidth on VCs such that RT constraints can be guaranteed. Finally, we use the ABR service on the VCs interconnecting the routers in Figure 2 where a Minimum Cell Rate (MCR) is allocated to implement bandwidth guarantees to the backbone traffic. This approach, as described in (Pazos *et al.* 1997), improves ATM network utilization and throughput for the Internet traffic as compared to the more common approach of using the CBR service.

## 4 FLOW CONTROL APPROACH

Let us now elaborate a little further on the congestion issue we are addressing in this paper. First of all, Vd sessions are likely to last for more than a few minutes, they reserve resources in the network and they are given priority over the Dt traffic by the CBQ scheduler. Hence, in this scenario if the Vd traffic is well behaved or policed, it is unaffected by congestion in the network. On the other hand, TCP sessions are likely to be responsible for the majority of the BE traffic. The TCP flow control tests for congestion in the network by increasingly sending more packets. Packets must then be dropped for the transmission window to be reduced and the source rate to be controlled. However, the dropping of packets is very undesirable because it can lead to global synchronization and throughput collapse, and it compromises the performance of delay sensitive traffic (e.g., telnet) (Floyd 1995, Morris 1997).

Approaches such as Random Early Detection (RED) (Floyd *et al.* 1993) and Explicit Congestion Notification (ECN) (Floyd 1995) try to make the packet dropping more “effective and fair” and to enhance the congestion feedback, respectively. RED uses queue length information to randomly decide when to drop packets. ECN improves on RED by marking the packets RED would normally drop as a means to indicate congestion to the sources. In any case, the source’s reaction to the congestion signal is felt at the congested router only after one RTT. Even then, the reduction in the source rate may not have been sufficient, in which case packets are still dropped and a new RTT has to go by before the rate is reduced again.

The equivalent of ECN has also been proposed in ATM under the name of FECN (Forward ECN) for the flow control of ABR connections (Newman 1994). In this case, the sources implement a multiplicative decrease of their sending rates upon congestion indication, and an additive increase when there is no congestion in the network. Such binary feedback may reduce implementation complexity, but it is well known to lead to unfairness, oscillation, and slow response to congestion (An *et al.* 1997). Hence, many ABR flow control schemes employ explicit rate indication (ATM Forum 1996, Jain *et al.* 1996) or relative rate indications (Chiussi *et al.* 1997), as a means to solve or at least alleviate these problems.

Challenges also arise in the transport of TCP traffic over ATM using the UBR service. The UBR traffic is not subjected to flow control at the ATM layer and cells are dropped when a congestion threshold is exceeded. In this scenario, the ATM switches usually employ Early Packet Discard (EPD) mechanisms (Romanow *et al.* 1994, Wu *et al.* 1997) to alleviate the source synchronization problem caused by tail dropping. Fairness problems also arise and selective dropping approaches such as the “Virtual Queueing” (Wu *et al.* 1997) address this issue. Since RED and ECN are also binary indication mechanisms which feedback congestion indications based on the state of a common FIFO queue, they are bound to experience similar fairness problems.

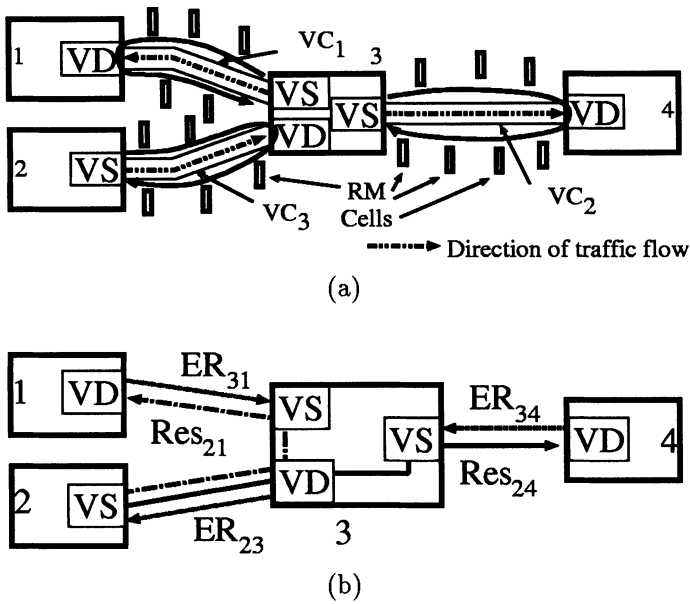
To sum up, the TCP flow control is reactive in nature with actions taken only after congestion sets in. Congestion notification based on packet drop is inadequate to preserve network throughput, to maintain fairness, or to deliver a quality service to delay sensitive traffic. Schemes are available to alleviate the consequences of dropping packets, but the response time is still in the order of RTTs. With high transmission rates becoming increasingly more common and with the number of simultaneous sessions rapidly increasing, such traditional approach may not react fast enough to alleviate congestion. In the next section we introduce a back-pressure approach with response time in the order of the RTT between the source and the point of congestion.

#### 4.1 Back-pressure Flow Control

Our goal here is to address the congestion problem of Figure 1 as discussed in section 2. However, from now on we will consider the more general scenario of Figure 2 in which router  $R_3$  can experience the congestion illustrated in Figure 1. The rationale of our approach is that since the VCs interconnecting the routers use the ABR service, the Internet traffic can make a better utilization of the available ATM resources (than with CBR) and still allow the routers to enforce a back-pressure flow control. In addition, routers at the end points of a VC function as virtual sources (VSs) and virtual destinations (VDs) for the aggregate traffic flowing between them through the VC. Figure 3(a) illustrates the VS and VD roles played by each router port as seen by the traffic flowing in the direction indicated.

Figure 3(a) illustrates how the ABR rate-based flow control is applied to each VC. A VS is continuously informed of the bandwidth available along its associated VC path through Explicit Rate (ER) indications on a stream of Resource Management (RM) cells (ATM Forum 1996). While ABR protects the VC path from congestion, it does not prevent congestion at the terminating router acting as VD. Hence, a back-pressure flow control can be implemented by having VDs reduce the ER indications on the RM cell stream, before sending them back to the VSs.

It is then critical to determine how to dynamically adjust the ER indications



**Figure 3** The ABR flow control applied to VCs.

and to assess the resulting impact on throughput, cell loss and fairness. Let us illustrate the issues involved by considering Figure 3(b). In this figure router 3 is an intermediary hop for the bidirectional traffic flowing between any two of the other routers shown. Let us further focus on the traffic from router 2 to routers 1 and 4. A portion of this traffic (RS) has made resource reservations  $Res_{21}$  and  $Res_{24}$  on the VCs traversed by the respective streams, as illustrated in Figure 3(b). These reservations are guaranteed by the MCR allocated to the VCs traversed by the RS flows and they are enforced by the CBQ scheduler on each output port. The traffic without reservation (NRS) is entitled to a fair share of the unused MCR bandwidth, the unassigned bandwidth and the bandwidth leftover by other ATM connection along the paths of the respective VCs. For the sake of argument, we further assume that  $ER_{31}$  and  $ER_{34}$  are the rates advertised by routers 1 and 4 and by the corresponding VCs. The issue is then to determine  $ER_{23}$  (i.e., the rate router 3 should advertise to router 2) such that the traffic from router 2 does not congest router 3.

In our approach, such back-pressure is actually provided as part of the VS and VD behaviors to be implemented by the ATM cards in the router ports. In (Pazos *et al.* 1997b), a VS and VD behavior was proposed for a simple tandem topology. Here we consider the more general topology of Figure 2 and we use the measured bandwidth utilization and the available bandwidth to slow down sending routers when their offered traffic would make downstream routers congested. In the following, we first disregard, for simplicity, the notion



of selective back-pressure and at the end of this section we discuss the resulting implications on throughput and fairness.

## 4.2 The Virtual Source and Destination Behaviors

For max-min fairness (Bertsekas *et al.* 1992), we measure the NRS traffic contribution from each input port and we allocate fair shares to input ports based on whether they have traffic to use the fair shares. In our studies, we consider a variation of the approach in (Bertsekas *et al.* 1992), which involves measuring used and available bandwidth. This does not necessarily lead to more complexity in our scenario since our CBQ scheduler knows how much bandwidth has been assigned to each class and it measures how much bandwidth each class is actually using in order to enforce the link sharing policy. A minor modification can then be made to allow the scheduler to keep track of bandwidth usage by individual input ports.

Note, however, that the CBQ queues contain packets. Determining the input port a packet comes from may not be possible given that it would involve the same amount of processing overhead as that needed to make packet routing decisions. In our studies, though, we make the input ports stamp their own local identification on packets, using a sort of internal encapsulation header, before sending them to the output port which then strips this extra header off before sending the packet onto the outgoing link (such approach is not uncommon on the architecture of some ATM switches).

The only real issue is that this bandwidth bookkeeping has to be done at the IP layer (we implement it as part of the CBQ scheduler code) and periodically communicated to the ATM layer, as described below. In our implementation, the CBQ scheduler on each output port keeps track of the traffic offered (in bytes) by each class and each input port  $i$  ( $Dt_i$  and  $Res_i$ ), and it keeps track of the traffic actually sent ( $Tf_{out}$ , also in bytes) through the output port during a fixed sampling period.  $Tf_{out}$  reflects the ER indication (the available bandwidth) for the outgoing VC, while  $MCR_{out}$  (in bytes) is the traffic that would be sent if  $ER = MCR$ . Finally,  $RESV_i = \alpha_i \times MCR_{out}$  reflects the bandwidth reserved for the traffic from input port  $i$ .

The Virtual Source behavior is carried out by the ATM layer and the CBQ scheduler in the following way:

1. At the beginning of a new sampling period:

- (a) The CBQ scheduler computes the fair share for the  $Dt$  class as:

$$Av_{Dt} = Tf_{out} - \sum_i^N RESV_i, \quad FS = \frac{Av_{Dt}}{N},$$

$Av_{Dt}$  is the available resources (in bytes) for the Dt traffic and  $N$  is the number of input ports contributing traffic to the output port.

- (b) If an input port offered traffic is such that  $Dt_i + Res_i < FS + RESV_i$ , the traffic from input port  $i$  is constrained somewhere else and the respective input port is marked as unconstrained.
- (c) The remaining input ports  $j$  are constrained on the router and they are assigned an actual share:

$$AS_j = \frac{Tf_{out} - \sum_i^{N_{un}} (Dt_i + Res_i) - \sum_k^{N-N_{un}} RESV_k}{N - N_{un}} + RESV_j$$

$N_{un}$  is the number of input ports unconstrained at the output port.

- (d) Less obvious is the possibility that  $Dt_i + Res_i > AS_i \times Tf_{out}$ . This means that input port  $i$  is actually sending too much traffic to the output port. Let  $Ex_i = Dt_i + Res_i - AS_i \times Tf_{out}$  be the excess traffic sent from input port  $i$ . Since this excess traffic on the previous sampling time was caused by reasons outside the scope of the output port, we expect at least another  $Ex_i$  in the next sampling period. Hence, we make:

$$AS_i^{(new)} = \frac{AS_i^{(old)} \times Tf_{out} - 2Ex_i}{Tf_{out}}.$$

- (e) If an input port is constrained somewhere else, assigning it its fair share does not congest the router while the input port is constrained. However, when the input port is no longer constrained somewhere else before getting to the router, this allows the corresponding traffic to reclaim its fair share. Hence, all unconstrained input ports are assigned an actual share:

$$AS_i = \frac{FS + RESV_i}{Tf_{out}}$$

- (f) Finally, the CBQ scheduler sends the vector AS to the ATM layer through a control primitive.

2. At the ATM layer, a flow of backward RM cells (ATM Forum 1996) is continuously being returned as part of the ABR flow control. The ER indication on these cells advertise the bandwidth actually available along the VC and supported by the router on the other end. Hence, upon the arrival of a new forward RM cell, the ATM layer uses the current vector AS and sends to the ATM interface on each input port  $i$  in the router a

control message with  $AS_i * RM.ER$  as the bandwidth available for the input port  $i$  to send traffic to the output port.

The Virtual Destination behavior carried out by the ATM layer on input ports is:

1. A vector  $Bw_i$  keeps track of the bandwidth available for the input port traffic on each other output port  $i$ , and an entry  $Bw_i$  is updated every time a new control message arrives from the output port  $i$ .
2. Whenever a forward RM cell arrives, we test if  $RM.ER$  is smaller than the sum of all  $Bw_i$ . If it is, the ER indication is left intact, otherwise it is replaced by the sum of all  $Bw_i$ .
3. The RM cell is sent back to the VS as a backward RM cell.

### 4.3 Max-min Fairness Considerations

In our simulation results we demonstrate the effectiveness of this approach to implement back-pressure and to achieve max-min fairness. Now, let us elaborate on the implications to the congestion control problem of Figure 1. If we apply the VS and VD behaviors above to the scenario of Figure 1, we would get for instance an  $ER_b$ , the bandwidth available to the router feeding link b. However this ER indication does not reflect  $AS_{bc}$  and  $AS_{bd}$ . Rather it reflects  $AS_{bc} + AS_{bd}$  and the sending router has no way of enforcing  $AS_{bc}$  and  $AS_{bd}$  selectively. If  $\lambda_{bc}$  and  $\lambda_{bd}$  are smaller than or equal to  $AS_{bc}$  and  $AS_{bd}$ , respectively, we have no problems.

However, if both flows  $\lambda_{bc}$  and  $\lambda_{bd}$  have enough traffic load, they would be assigned half of  $AS_{bc} + AS_{bd}$ , which is clearly a problem if  $AS_{bc} \neq AS_{bd}$ . In this case, it is easy to see that one link will get congested while the other will be under utilized. However, with appropriate buffering, we can accommodate the excess traffic and, from the VS behavior action 1(d), we try to alleviate this effect by reducing the actual share for the input port creating congestion.

### 4.4 Explicit Rate and the Last Hop

As mentioned earlier, the Explicit Rate indications provided by ABR cannot be communicated to the TCP sources without ATM connectivity "to the desk top". The issue is then the usefulness of the ER indications if the actual sources do not receive this information. This problem was already addressed in (Kalyanaraman *et al.* 1996, Narcáez *et al.* 1997). Since, with the arrival of acknowledgments, the TCP sources can double their transmission window and send more packets, and since the sources connected to a Legacy LAN are

not subjected to any flow control other than TCP, the traffic will reach the edge router at a rate higher than that supported by the ATM network.

If the number of active TCP sessions is high, one expects that the edge router may run out of buffers and may start dropping packets. On the positive side, no network resource is consumed by packets that would be eventually dropped inside the network. An alternative approach proposed in (Kalyanaraman *et al.* 1996) consists on equipping edge devices with enough buffers to hold many TCP-receiver-windows worth of packets.

A more interesting idea was presented in (Narcáez *et al.* 1997) as a means of “effectively extending the ABR flow control all the way to the TCP source”. In their approach the edge router uses an acknowledgment bucket that translates the ER indications into a sequence of TCP acknowledgments to prevent the TCP source from sending at rates higher than the ER indications. A potential difficulty with this approach is that the TCP acknowledgments may in fact be returned piggy-backed on the packets from the reverse data flow. Thus, implementing this approach may require altering the acknowledgment field in the TCP header.

Another approach would use ECN (Floyd 1995). Again, with the arrival of acknowledgments the sources would increase their transmission windows and the buffers on the edge routers would fill up. Hence, using ECN in the **forward** direction has the same problems as discussed in section 4. However, we can stamp the ECN bit on acknowledgment packets **returning** through the edge device.

## 5 SIMULATION RESULTS

In this section we present simulation results to illustrate the effectiveness of our back-pressure approach. For our study, we used the topology illustrated in Figure 2, in which all links are 150Mbps, the propagation delay on the indicated VCs is 400 $\mu$ s, and the propagation delay from a host to an edge router is 10 $\mu$ s. Each Vd stream is composed of seven individual H.261 video (Turletti *et al.* 1996) flows transported over UDP at a target rate of 1.5Mbps and with 15 consecutive H.261 frames sent on every IP packet. Each Dt stream is made up of ten individual ftp session transferring large files (persistent sources). The routers have 100-packet and 30-packet queues for the aggregate Dt and Vd streams, respectively. Our simulator code implements the TCP Tahoe version and we consider a maximum segment size of 1024 bytes.

Furthermore, each VC in Figure 2 traverses two ATM switches (not shown to avoid overloading the figure) and one of the links traversed by each VC is shared by other CBR traffic. We chose the load for the background CBR traffic such that the residual bandwidth  $C_{ij}$  available along the  $VC_{ij}$  paths in Figure 2 is 70 Mbps, except for  $C_{23}$  and  $C_{63}$  which are 95Mbps. The  $MCR_{ij}$  allocated to the  $VC_{ij}$ s terminating in  $R_3$  is 40Mbps, the others are allocated an MRC of 20Mbps. In our experiments, we modify the CBR load to test

different congestion scenarios. Finally, the  $MCR_{ij}$  for a VC is taken by the CBQ scheduler to be the link bandwidth for the purpose of imposing the link sharing policy of 70% of MCR dedicated to the video traffic, and 30% dedicated to the data traffic. The data traffic can of course use all of the bandwidth actually available on the links.

## 5.1 Using Back-pressure

We study the effectiveness of our scheme using the traces in Figures 4 and 5. These traces show the Dt, Vd and Total load offered to the VC connected to the routers indicated. Figure 4 plots the traces for the core routers, while Figure 5 plots the traces for the edge routers of Figure 2. All connections start at different times, but the actual starting time is uniformly distributed over an initial time interval.

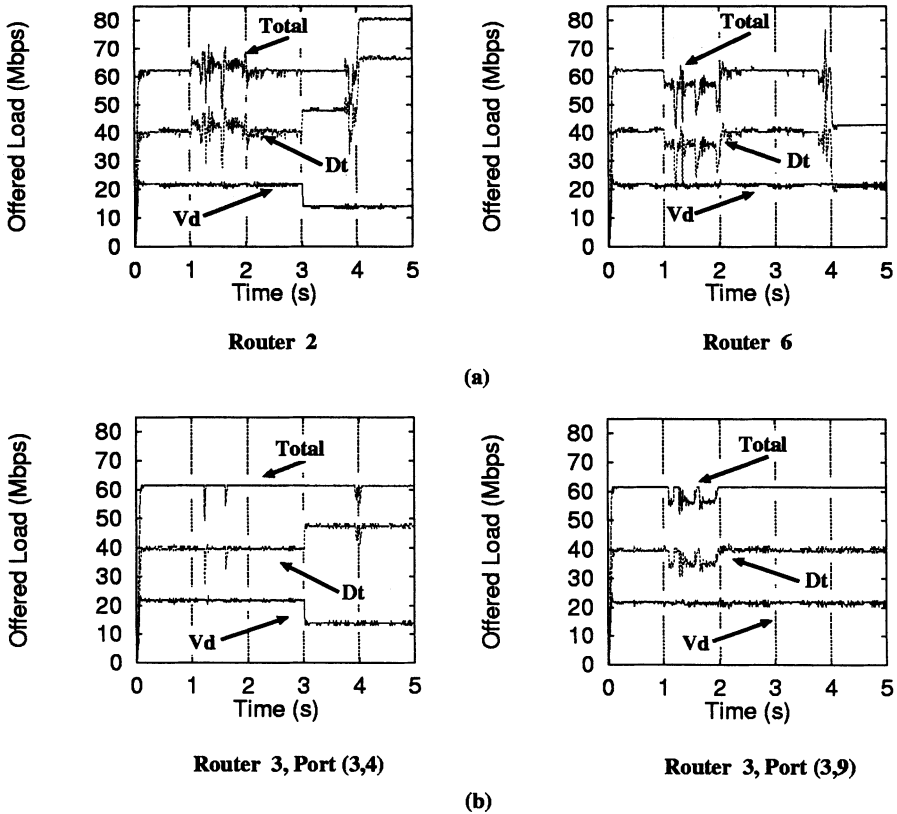
### (a) A Load Balanced Scenario

We study a different load scenario in each of the five seconds of Figures 4 and 5. So, **before  $t = 1s$** , we have all connections active, and the initial available bandwidth on the links are the ones indicated above. However, as the traces in Figure 4(b) indicate, the actual rate available on the respective links is roughly 60Mbps, not the 70Mbps mentioned above. This is so because we plot the effective rate, discounting all the different protocol (UDP, TCP, IP and ATM) overheads. Note that 60Mbps is also the actual rate at which  $R_2$  and  $R_6$  transmit to  $R_3$ , even though the VCs connecting them to  $R_3$  have 95Mbps of available bandwidth. This means that the VD entities on  $R_3$  are appropriately reducing the ER indication they receive on forward RM cells, before returning them to the VS entities in the sending routers. Hence,  $R_3$  is effectively back-pressuring  $R_2$  and  $R_6$ .

If we now look at the traces in Figure 5, we see that each edge router is roughly transmitting at a maximum aggregate rate of 30Mbps, when the bandwidth on the VCs connecting the edge routers to the core routers  $R_2$  and  $R_6$  is actually 70Mbps. The reason is that the traffic from  $R_1$  and  $R_5$  and from  $R_7$  and  $R_8$ , must share VCs with 60Mbps bandwidth constraints from routers  $R_2$  and  $R_6$  to router  $R_3$ , respectively. With this, we observe that the core routers effectively back-pressure the edged routers.

### (b) The Effect of Unbalanced Load

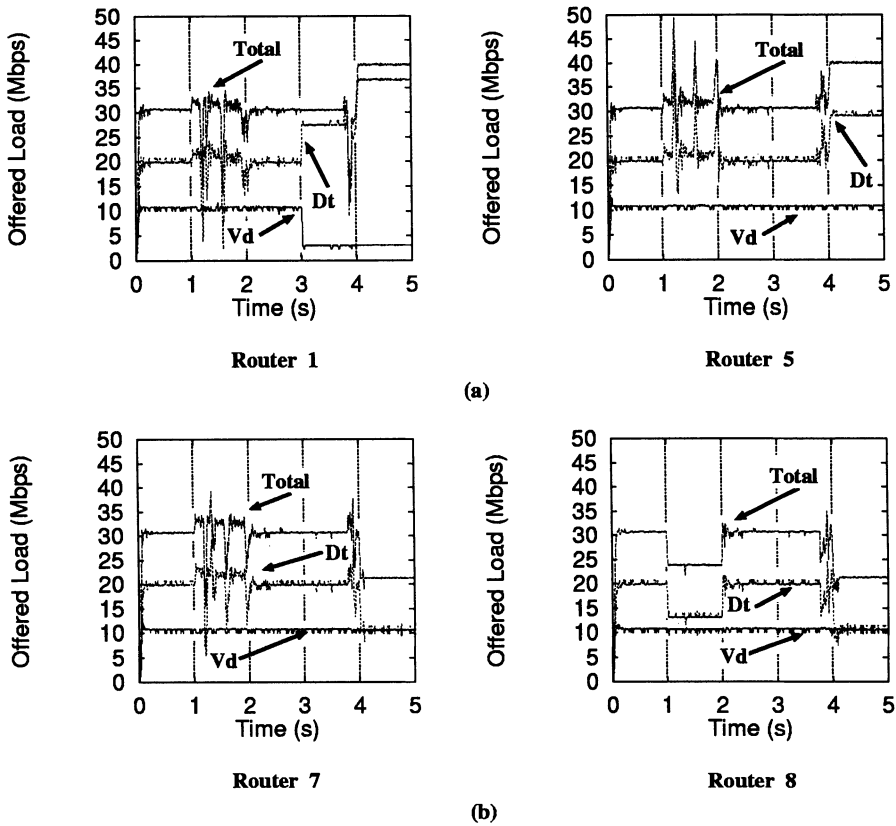
**At time  $t = 1s$** , we simulate the scenario in which the CBR traffic is increased on the trunk (8,6) thus reducing  $C_{86}$  to roughly 30Mbps. As Figure 5(b) shows, the Dt traffic through  $R_8$  is reduced accordingly while the Vd traffic is unaffected since we have the CBQ scheduler. Moreover, for  $R_6$ , the traffic stream fed by  $R_8$  becomes constrained somewhere else. Thus the VS on  $R_6$



**Figure 4** The load offered by core routers.

increases the Actual Share (AS) of the  $VC_{63}$  bandwidth to the input port fed by  $R_7$ . This is consistent with the max-min fairness policy. However, the bandwidth left over by the Dt traffic from  $R_8$  cannot be claimed by the Dt from  $R_7$  because, as Figure 2 indicates, these streams only share  $VC_{63}$ . The Dt from  $R_7$  is still constrained by the bandwidth available on  $VC_{34}$ . Hence, some of the excess packets are unavoidably lost on  $R_3$ .

Then, after the reduction of the traffic load from  $R_8$  is felt on  $R_3$ , the VS acting on  $VC_{39}$  deems  $VC_{63}$  constrained somewhere else and it increases the Actual Share for  $R_2$ , again in an attempt to implement max-min fairness. However, the newly available bandwidth is split in  $R_2$  between the traffic from  $R_1$  and  $R_5$ . The traffic from  $R_5$  could recover the bandwidth released by  $R_8$ , but we cannot enforce this in  $R_2$  and the result are the oscillation we see in the interval [1,2]. The excess traffic sent from  $R_5$  goes through unharmed



**Figure 5** The load offered by edge routers.

while the excess traffic from  $R_1$  collides with the excess traffic from  $R_7$  and losses cannot be prevented if the scenario persists for a long time.

### (c) Using the Allocated Fair Shares

At time  $t = 2s$ , the CBR extra load on trunk (8,6) is removed,  $C_{86}$  returns to its original value and the Dt traffic from  $R_8$  recovers the bandwidth that is available along the path to  $R_9$ . This is due to the fact that the  $R_3$  output port (3,9) regarded the input port (6,3) constrained but it still allocated this input port its fair share (see VS behavior 1(e)). So, when the  $R_8$  traffic started using its fair share, the AS for input port (2,3) is also brought back to the fair share and the oscillations present in the interval [1,2] are terminated.

At time  $t = 3s$  we illustrated another type of traffic load variation: the aggregate Vd traffic load offered through  $R_1$  is reduced. This simulates the

case in which some of the Vd sessions are closed. However, this does not reduce the reservations that are made on CBQ schedulers for the Vd flows. As a result, the Dt traffic from  $R_1$  benefits from the bandwidth released by the Vd sessions.

#### (d) Testing Max-min Fairness

Finally, at time  $t = 4s$  another increase in the CBR load reduces now the bandwidth  $C_{63}$  to 50Mbps. As Figure 4(a) illustrates, the Dt traffic from  $R_6$  is reduced accordingly. With the  $VC_{63}$  constrained by the excess CBR traffic, the BE traffic from  $R_7$  and  $R_8$  are also reduced in response to the back-pressure from the VS in  $R_6$ . The Dt reduction affects the loads on  $VC_{34}$  and  $VC_{39}$ . In particular, the respective output ports regard the input port (6,3) constrained and they increase the AS for input port (2,3). This extra bandwidth is then notified by the back-pressure mechanism to  $R_1$  and  $R_5$  which can use the extra bandwidth. Hence their load is adjusted accordingly as illustrated in Figure 4(a), for core router  $R_2$ , and in Figure 5(a), for edge routers  $R_1$  and  $R_5$ .

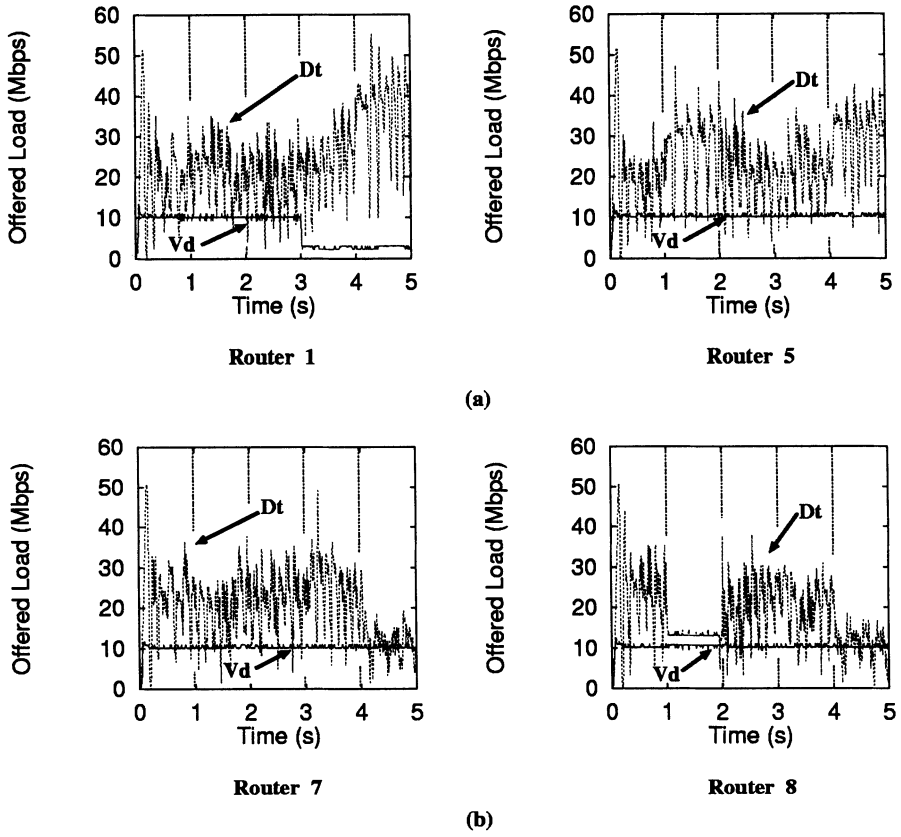
## 5.2 Comparing Results With and Without Back-pressure

The simulation results presented so far were designed to illustrate some of the features as well as limitations of our back-pressure control. Next, we compare these results with those of a network without back-pressure. For the latter case, we present the simulation results in Figure 6 for edge routers only. First of all, the TCP sources become active with a slow-start and the offered traffic increases exponentially (doubling every RTT). Since we no longer employ the back-pressure, this traffic is always admitted into the network. Hence, the aggregate offered traffic achieves a 50Mbps peak after  $t = 0s$  and many packets are dropped.

Comparing these results with the results in Figure 5, it is clear that back-pressure is effective in avoiding these losses. Furthermore, since TCP sources continuously increase their transmission windows, packets are periodically lost and the windows shrink to  $W = 1$ . This accounts for the oscillations throughout the simulated time. Since these losses are prevented by our back-pressure, the traces in Figure 5 are smoother.

At  $t = 1s$ , the available rate on  $VC_{86}$  is reduced. This affects only the traffic from  $R_8$ , which drops many packets and keeps a persistent backlog allowing full utilization of the available bandwidth on  $VC_{86}$  (Figure6(b)). Note, though, that there is no side effects on the traffic offered by  $R_1$  and  $R_7$  through  $t = 3s$  since their respective flows do not share a common path with traffic from  $R_8$ . The traffic from  $R_5$ , on the contrary, can claim the bandwidth leftover. This is the type of scenario in which the back-pressure would affect the offered traffic from sessions not affected by the congestion.





**Figure 6** The load offered by edge routers under no back-pressure.

At  $t = 2s$ , the available rate on  $VC_{86}$  is increased again and the backlog in  $R_8$  is dumped into  $R_6$ , causing buffer overflow in  $R_3$ . Here again, in the back-pressure scenario of Figure 5, the new offered load reclaims its fair share while the load from  $R_5$  gets back-pressured and these losses are avoided. Finally, note that in the interval  $[2,3]$  the bottleneck for all connections are  $VC_{34}$  and  $VC_{39}$ . But at  $t = 3s$ , the reduction on the Vd load through  $R_1$  allows the Dt streams from  $R_1$  and  $R_7$  to claim the leftover bandwidth. However, since the Dt stream from  $R_8$  cannot claim this newly available bandwidth, this traffic suffers with the extra traffic sent from  $R_7$ . With the use of back-pressure this is avoided.

### 5.3 Other Performance Metrics

In Table 1 we present the number of packets lost and the aggregate throughput accumulated over the five-second interval considered in the above study. As these results show, the use of back-pressure is effective in reducing the number of packets lost. The throughput performance, on the other hand, is almost identical. The reason for this is that the number of active TCP sessions is large compared to the buffering and delay-bandwidth product. This is an indication that in the presence of a vary large number of sessions, it is not likely that all of them will be periodically synchronized leading to the loss of TCP throughput, which RED and ECN try to avoid. However, average TCP throughput alone does not tell the whole story. The bursty behavior observed in Figure 6 implies considerable fluctuations in TCP end-to-end delays over individual sessions, which is a serious performance problem for delay sensitive applications such as telnet and web browsing. The study of these effects is left for future research.

**Table 1** Packet loss and aggregate throughput.

Metric	No back-pressure	Back-pressure
Packet Losses	1957	135
Aggregate Throughput	79.27Mbps	79.66Mbps

## 6 CONCLUSIONS AND FUTURE RESEARCH

In this paper we address the effectiveness of the TCP flow control to deal with large delay-bandwidth products and with very many simultaneous connections (Morris 1997) observed in Internet backbones. Since ATM has been extensively used as a link layer for Internet backbones, and this trend is likely to persist for the near future, we address this congestion control problem in the IP over ATM scenario. Our approach follows the suggestions in (Pazos *et al.* 1997) to use the ABR service in the backbones and we define an appropriate Virtual Source and Virtual Destination behavior for the IP routers terminating the ABR control loop. Such approach effectively implements a back-pressure mechanism that addresses max-min fairness. In addition, it is also more effective in handling the large RTTs and the large number of TCP sessions because we back-pressure from the point of congestion to the sources and because we back-pressure the aggregate traffic between routers.

As for future research, we are looking into the use of Label Swapping Routers (LSR) over ATM to implement this back-pressure mechanism. The

use of LSR should lead to a more elegant mechanism because all queueing, scheduling and bandwidth bookkeeping can be done entirely at the ATM layer. This removes the need for passing control messages across layer boundaries as described in section 4.2 as part of the Virtual Source behavior.

## ACKNOWLEDGMENTS

This research was partially funded by GTE, NSF and CNPq.

## REFERENCES

- Braden, R. and Clark, D. and Shenker, S. (1994) Integrated Services in the Internet Architecture: an Overview. *Request for Comments 1633*.
- Morris, R. (1997) TCP Behavior with Many Flows. *Proc. of ICNP '97*.
- Pazos, C. M. and Gerla, M. (1997) ATM Virtual Private Networks for the Internet Data Traffic. *Proc. of MMNS '97*.
- ATM Forum Technical Committee (1996) Traffic Management Specifications, Version 4.0.
- Paxson, V. (1993) Empirically-Derived Analytic Models of Wide-Area TCP Connections: Extended Report. *Lawrence Berkeley Laboratory Technical Report*.
- Jain, R. and Kalyanaraman, S. and Goyal, R. and Fahmy, S. and Viswanathan, R. (1996) ERICA Switch Algorithm: A Complete Description. *ATM Forum Contribution. AF-TM 96-1172*.
- Floyd, S. (1995) TCP and Explicit Congestion Notification. *ACM Computer Communication Review*, **24**(4).
- Roche, C. and Plotkin, N. (1995) The Converging Flows Problem: an Analytical Study. *Proc. of INFOCOM '95*.
- Floyd, S. and Jacobson, V. (1991) Traffic Phase Effects in Packet-Switched Gateways. *ACM Computer Communication Review*, **21**(2).
- Floyd, S. (1991) Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1: One-way Traffic. *ACM Computer Communication Review*, **21**(5).
- Floyd, S. and Jacobsen, V. (1995) Link-sharing and Resource Management Models for Packet Networks. *IEEE/ACM Transactions on Networking*, **3**(4).
- Floyd, S. and Jacobsen, V. (1993) Random Early Detection gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, **1**(4).
- Newman, P. (1994) Traffic Management for ATM Local Area Networks. *IEEE Communications Magazine*, **32**(8).
- An, L. and Ansari, N. and Arulambalam, A. (1997) TCP/IP Traffic over ATM Networks with ABR Flow and Congestion Control. *Proc. of GLOBE-COM '97*.

- Chiussi, F. and Xia, Y. and Kumar, V. (1997) Virtual Queueing Techniques for ABR Service: Improving ABR/VBR Interaction. *Proc. of INFOCOM '97*.
- Romanow, A. and Floyd, S. (1994) Dynamic of TCP Traffic Over ATM Networks. *Proc. of SIGCOMM '94*.
- Wu, Y. and Siu, K-Y. and Ren, W. (1997) Improved Virtual Queueing and Dynamic EPD Techniques for TCP over ATM. *Proc. of ICNP '97*.
- Pazos, C. M. and Gerla, M. (1997b) Bandwidth Efficiency in Internet AVPNs. *Proc. of GLOBECOM '97*.
- Bertsekas, D. and Gallager, R. (1992) Data Networks. *Prentice Hall*.
- Kalyanaraman, S. and Jain, R. and Fahmy, S. and Goyal, R. and Jiang, J. (1997) Performance of TCP over ABR on ATM Backbone and with Various VBR Traffic Patters. *ATM Forum Contribution 96-1294*.
- Narcáez, P. and Siu, K-Y. (1997) An Acknowledgment Bucket Scheme for Regulating TCP Flows over ATM. *Proc. of GLOBECOM '97*.
- Turletti, T. and Huitema, C. (1996) RTP Payload Format for H.261 Video Streams. *Request for Comments 2032*.

## BIOGRAPHY

**Carlos Marcelo Dias Pazos** received his graduate degree in Electrical Engineering and his M. S. degree in Computer Science from Federal University of Pernambuco, Brazil in 1990 and 1993, respectively. Since 1993, he has been a Ph. D. student with the Computer Science Department at University of California, Los Angeles. His research interests include high speed communication systems and networks, ATM networks, multi-service Internet, and Internet over ATM.

**Dr. Mario Gerla** received his graduate degree in Electrical Engineering from Politecnico di Milano, Italy in 1966 and his M. S. and Ph. D. degrees in Computer Science from University of California, Los Angeles in 1970 and 1973, respectively. ¿From 1973 to 1976, Dr. Gerla was a manager in Network Analysis Corporation, Glen Cove, NY, where he was involved in several computer network design projects for both government and industry, including performance analysis and topological updating of the ARPANET under a contract from DoD. From 1976 to 1977, he was with Tran Telecommunication, Los Angeles, CA, where he participated in the development of an integrated packet and circuit network. Since 1977, he has been on the Faculty of the Computer Science Department of UCLA. His research interests include the design, performance evaluation, and control of distributed computer communication systems and networks. His current research projects cover the following areas: topology design and bandwidth allocation in ATM networks; design and implementation of optical interconnects for supercomputer applications, and; network protocol design and implementation for a mobile, integrated services wireless radio network.

# TCP-BFA: Buffer Fill Avoidance

*A. A. Awadallah, C. Rai*

*Computer Systems Laboratory, Stanford University*

*Gates 3A, Stanford University, Stanford CA 94305, USA, Tel: +1-650-723-1414, Fax: +1-650-725-6949, {aaa,crai}@cs.stanford.edu*

## Abstract

The main goal of a congestion avoidance algorithm is to maximize throughput and minimize delay (Jain & Ramakrishnan 1988). While TCP Reno achieves high throughput, it tends to consume all of the buffer space at the bottleneck router, causing large delays. In this paper we propose a simple scheme that modifies TCP Reno's congestion avoidance algorithm by throttling back the opening of the congestion window once an increase in round-trip time is perceived. We call the scheme TCP-BFA and have implemented it in the *ns* network simulator and in BSD 4.4. We show through simulations and measurements of real traffic on the Internet that TCP-BFA results in lower router buffer occupancies and lower delays while maintaining a throughput similar to that of TCP Reno. The advantages of TCP-BFA are (1) smaller router buffer size requirements, (2) an order of magnitude improvement in network power (the ratio of throughput to delay), (3) fewer packet losses, (4) faster detection of multiple losses due to lower retransmission timeout estimates, and (5) smoother traffic patterns.

## Keywords

TCP, congestion control, congestion avoidance, Internet, transport protocols, optimal window

## 1 INTRODUCTION

TCP is the most widely used transport protocol for today's Internet data applications. However, the performance and operation of TCP's adaptive retransmission and congestion control mechanism is one of the most widely debated issues in the research community. Revisions for TCP have been proposed over the years (e.g. Jacobson 1988, Jacobson 1990, Jacobson, Braden & Borman 1992, Brakmo, O'Malley & Peterson 1994, Hoe 1996, Mathis, Mahdavi, Floyd & Romanow 1996, Mathis & Mahdavi 1996, Wang & Crowcroft 1991, Wang & Crowcroft 1992, Floyd 1995), with Jacobson's paper (Jacobson 1988) representing a major milestone. Some of these proposed changes have been widely adopted and are part of TCP implementations today.

The source has the primary responsibility for TCP's congestion avoidance

and control. The sink simply sends ACKs back to the source for the data it receives; no explicit congestion control information is sent back. The network also provides no explicit notification about its state to the source (Source Quench messages are rarely used, and Floyd's (1995) Explicit Congestion Notification is yet to be deployed), further complicating the task of the source. The source needs to make control decisions about the network by guessing the state of the network from the information it has: the ACKs it receives from the sink, and the timing information it obtains by estimating the round trip time (RTT) of the transmitted packets.

In the absence of explicit feedback from the network to the source, schemes in two flavors have been proposed to provide high throughput with low delay. The first kind, such as Tri-S (Wang & Crowcroft 1991), RED gateways (Floyd & Jacobson 1993) and SACK (Mathis et al. 1996) (with associated algorithms such as FACK (Mathis & Mahdavi 1996)), extend the functionality of routers or TCP sinks to provide this missing feedback. Others, such as CARD (Congestion Avoidance using Round-trip Delay) (Jain 1989), Keshav (1991), the DUAL algorithm (Wang & Crowcroft 1992), TCP Vegas (Brakmo et al. 1994) and Hoe (1996), focus on the transport-layer protocol at the source. For example, Vegas allows TCP to stop below the point where it would lose packets, hence providing a smoother stream. Ahn, Danzig, Liu & Yan (1995) show that Vegas achieves from 3 to 8% better throughput, with only 1/5 to 1/2 of the losses, as compared to the BSD-Reno distribution, though the stability of Vegas has been questioned by Jacobson (1994).

Reno sources exhibit a start-stop behavior because they increase their window sizes, filling up buffers in the network and consequently suffering a packet loss or a timeout and falling back to a smaller window size. No attempt is made to predict the optimal window size: the size at which the (bandwidth-delay product) pipe is full and the buffers are empty.

In this paper we propose a method to improve congestion control in TCP without requiring any additional explicit information from the sink or the network. Our goals are to

- make TCP seek the optimal window size and operate close to it. This leads to
  - lower delay, which results in higher network power\* and faster recovery from multiple losses.
  - lower buffer occupancies, which allows routers to support a higher load for a given buffer size; conversely, a smaller buffer can be used to support the same load (without a corresponding increase in the number of packet losses).
  - a reduction in fluctuations in the round-trip delay and the number of

---

\*Network power is defined as the ratio of throughput to delay (Jain 1989, Kleinrock 1979).

packets in the network, allowing better interaction with non-TCP traffic such as real-time streams.

- only make modifications that are simple enough to incorporate into current TCP implementations.

TCP-BFA is similar to Reno. It realizes the above goals primarily by freezing the congestion window on a sustained increase in RTT. Upon detecting a sustained decrease in RTT, TCP-BFA reverts to normal Reno behavior. Hysteresis is used to avoid rapid flipping between the two states. To detect sustained changes in RTT, TCP-BFA uses an extra state variable that maintains the *signed* RTT variance. Since TCP-BFA relies heavily on RTT estimates, a timer granularity finer than the usual 500ms is required. It should be stressed, however, that this finer granularity is used only for RTT estimates; *the retransmission timeout (RTO) is still calculated with the coarse 500ms granularity for stability reasons* (Jacobson 1994). Since TCP-BFA has the additive increase and multiplicative decrease properties of Reno, with the change that additive increase is sometimes stopped, it is stable\* if Reno is stable.

Our simulations show that TCP-BFA requires considerably less buffer space in routers to attain the same throughput as Reno. Both the simulations and the Internet measurements show that TCP-BFA achieves smaller RTT averages – thus up to an order of magnitude higher network power – than Reno. When TCP-BFA shares the bottleneck link with one or more Reno sources, the gains in network power are modest to nonexistent since the Reno sources force higher buffer occupancies. In this situation, simulations indicate that TCP-BFA achieves a higher throughput than Reno, while the measurements show comparable values. It might appear that TCP-BFA will *always* be in competition with Reno, but we can expect a single form of TCP on a large number of paths with the bottleneck at the tail link (e.g., home PCs behind low speed modems).

The proposed algorithm and the concepts behind it are discussed in Section 2. In Section 3 we provide a simple implementation which is a heuristic approximation to the discussed algorithm. Sections 4 and 5 report simulation results and Internet measurements, respectively. Section 6 concludes the paper.

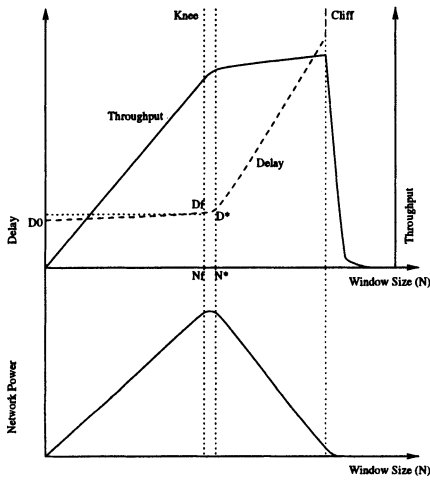
## 2 OPTIMAL WINDOW SIZE

Consider a source that uses window-based flow control connected to a sink through a number of links and buffers with out-of-band ACKs. Figure 1 (from (Jain 1989)) shows typical throughput, delay and power curves plotted against

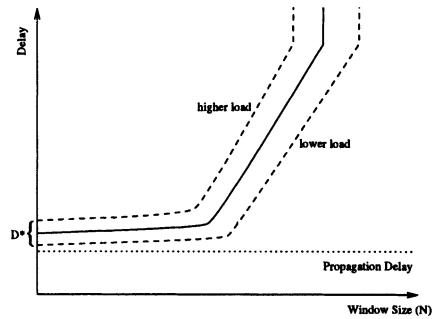
---

\*By stable, we mean that it does not fall off the cliff of Figure 1 (see Section 2).

window size for this scenario. An efficient window flow control protocol should operate at the point  $N^*$  where the throughput reaches a plateau and the delay starts increasing. Increasing the window size  $N$  beyond this point will not significantly improve the throughput but will increase the delay. Note that  $N^*$  lies on the knee of the delay curve. It can be shown (Fendick, Mitra, Mitrani, Rodriguez, Seery & Weiss 1991) that the value of  $N^*$  is very close, but slightly higher, than the value  $N_f$ , where  $N_f$  is the number of packets that can be stored in the bandwidth-delay product pipe between the source and the sink (with zero occupancy of the buffers). Basically, we need  $N^* > N_f$  to account for the stochastic nature of the traffic arrival pattern: this ensures that the bottleneck link remains fully utilized.



**Figure 1** Window flow control: the optimal window size. The plots are for a single source running on an unloaded network.



**Figure 2** Window flow control: dynamic network behavior. The plots are for a source running on a loaded network.

If the source starts with a window size of one it will experience the smallest RTT ( $D_0$  in the figure). As the window size increases, the RTT will stay constant at  $D_0$  ( $\approx D_f \approx D^*$ ) until the source reaches the point where it starts filling buffers and increasing the RTT (this happens when it reaches  $N^*$ ). Now,

$$\begin{aligned} \text{for } N < N^*, \quad \text{RTT} &\approx D^* = N^*/\rho, \\ \text{for } N > N^*, \quad \text{RTT} &= N/\rho, \end{aligned}$$

where  $\rho$  is the rate at which the ACKs arrive at the source (i.e., our share of the bottleneck link bandwidth). Note that for  $N < N^*$ , the window is not big enough to cover the RTT and silence periods are introduced, leading to



underutilization of the network. On the other hand, for  $N > N^*$ , packets accumulate at the buffers and  $RTT > D^*$ .

Figure 2 illustrates the effect on delay caused by varying network load. We see that the points  $D^*$ ,  $N^*$ , and the slope vary with the load. The variation of  $D^*$  may not be obvious: it can occur due to routing changes, and/or constant background load generated by open-loop sources (such as multicast or streamed traffic).

The source will have a set of curves depending on the network load and network events (such as failing links, routing changes, etc.), but in all cases the following algorithm\* will track the optimal window size  $N^*$ :

```
if (  $\partial RTT / \partial N > \epsilon$  ) decrease  $N$ 
else increase  $N$ 
```

where  $\partial RTT / \partial N$  is the partial derivative of delay with respect to window size, and  $\epsilon$  is a small relative threshold value close to zero.

In practice, estimating  $\partial RTT / \partial N$  is difficult, given the dynamic nature of the network and its traffic. However, what we need to estimate is the sign of this quantity rather than the magnitude. Observing the fact that, for TCP,  $N$  is always increasing (except during packet losses, which can be handled separately), we can simplify the problem further to the estimation of the sign of  $\partial RTT$ . This simplification is the essence of our implementation, as described in the following section.

Previous work on tracking the optimal window size has used a variety of metrics: Jain (1989) proposes an algorithm (CARD) based on maximizing network power (see Figure 1); Wang & Crowcroft (1991) use the derivative of sending rate with respect to window size  $\partial \rho / \partial N$  (Tri-S); and Brakmo et al. (1994) use the difference between the actual throughput and an expected value (Vegas).

### 3 IMPLEMENTATION

We now introduce a simple heuristic approach that constrains TCP to operate close to the (continuously varying)  $N^*$ . Our changes are easy to integrate with current TCP implementations.

#### 3.1 Timer Granularity

Current TCP implementations use RTT estimates only for the calculation of RTO. Our modified implementation uses RTT estimates for congestion control, and the 500ms timer granularity is too coarse for this purpose. We

---

\*This follows from (Jain 1989).

increased the granularity with which RTT estimates are measured to  $10ms$ , while maintaining the same coarse granularity for RTO estimates. Achieving this finer granularity is straightforward and does not involve any changes in the transmit timer function of TCP; we simply redefined the meaning of a timer tick.

### 3.2 Signed RTT Variance

We maintain a signed RTT variance\* as an extra TCP state variable; this is our approximation to  $\partial RTT$ . We cannot use the (unsigned) RTT variance already maintained by TCP because it measures absolute differences. The signed RTT variance uses non-absolute differences and a filter gain constant ( $\alpha_{srv}$ ) of  $\frac{1}{2}$  instead of the  $\frac{3}{4}$  used by the unsigned RTT variance of (Jacobson 1988); its computation is otherwise identical to that of the RTT variance already present in Reno implementations. The gain constant of  $\frac{1}{2}$  was determined by experiment to provide a good balance between minimizing the error due to randomness and maximizing reactivity to network changes. Using the signed RTT variance instead of variables involving the (current) measured RTT has the effect of filtering out transient network noise. Note that use of the times-tamp TCP option improves the accuracy of the signed RTT variance because RTT samples are more frequent.

### 3.3 Buffer Fill Avoidance

We have introduced an additional TCP variable (BFA Flag) to define a new state, Buffer Fill Avoidance (BFA), in which the congestion window (cwnd) is held constant.

This state is entered, either from Slow Start or from Congestion Avoidance, whenever the signed RTT variance indicates a positive drift in RTT. The BFA state is left when there is a negative drift. To prevent rapid oscillation between states, we introduce hysteresis using separate thresholds for setting and clearing the flag. The BFA Flag is set as follows:

```

if ( BFA Flag is on )
    if (Signed RTT Variance <=  $\sigma_{off}$ )
        switch off BFA Flag
else
    if (Signed RTT Variance >  $\sigma_{on}$ )
        switch on BFA Flag

```

---

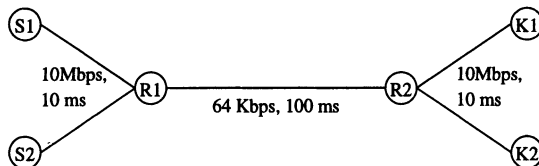
\*Let  $srv$  = signed RTT variance,  $\alpha_{srv}$  = filter gain constant, and  $\delta$  = measured RTT - smoothed RTT. Then,  $srv_{t+1} = \alpha_{srv} \cdot srv_t + (1 - \alpha_{srv}) \cdot \delta$ .

where  $\sigma_{\text{off}}$  and  $\sigma_{\text{on}}$  are the switch-off and switch-on thresholds, respectively ( $\sigma_{\text{off}} < \sigma_{\text{on}}$  for hysteresis). In addition, the BFA Flag is cleared whenever the congestion window drops due to a timeout or a fast retransmit.

If the threshold values are not properly tuned, the source can make false inferences about the state of the network. For instance, if the BFA Flag is set and the network load is constant, the signed variance tends to zero. If  $\sigma_{\text{off}}$  is non-negative, this will cause the BFA Flag to be switched off in spite of the constant load, which is clearly incorrect. Similarly,  $\sigma_{\text{on}}$  needs to be constrained to positive values. By adjusting the values of  $\sigma_{\text{off}}$  and  $\sigma_{\text{on}}$  the source can be tuned from ‘very aggressive’ to ‘extremely well-behaved’. We have used  $\sigma_{\text{off}} = -10\text{ms}$  and  $\sigma_{\text{on}} = 10\text{ms}$ .

As described in Section 2, a more proactive scheme would decrease the congestion window in an attempt to track the optimal value. We have chosen the simpler scheme of fixing the window and relying on normal TCP mechanisms for the decrease. The reason for this choice is that decreasing the window size makes the source much less aggressive, which causes it to yield bandwidth to Reno sources.

## 4 SIMULATION RESULTS



**Figure 3** Simulation Scenario

The results presented here have been obtained from the network simulator *ns 2.1* (McCanne & Floyd 1997) developed at LBNL.\* The scenario used for most of the simulations is shown in Figure 3. It models network paths with a single bottleneck link. This includes paths with a tail bottleneck (in the case of a machine behind a slow modem link) since the links between  $S^*$  and  $R1$ , and those between  $R2$  and  $K^*$ , never buffer packets. In fact, simulations with the bottleneck between  $R2$  and any of the  $K^*$  yield the same behavior.

All routers use tail-drop buffers. The bottleneck link bandwidth of 64 Kbps represents link speeds of today’s modems, and has the excellent side-effect of producing trace files of reasonable size! Simulations with higher bandwidths (such as 1.5 Mbps) gave similar results. The receiver window is set to a large value so that it does not limit the increase in the size of the congestion window.

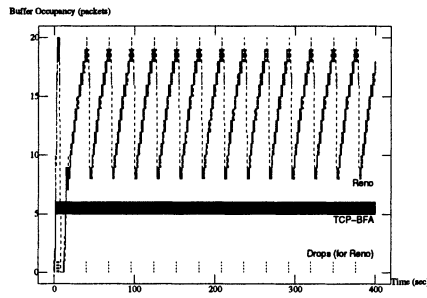
---

\*The *ns 2.1* implementation of TCP-BFA can be downloaded from <http://klamath.stanford.edu/~aaa/tcp-bfa>.

The TCP agent uses a packet size of 1000 bytes (64 Kbps translates to 8.0 packets/sec). All transfers are FTP bulk data transfers. Buffer size is measured in packets, and throughput in packets/sec.

#### 4.1 Single Sources

A typical queue behavior for Reno and TCP-BFA running separately on completely unloaded networks is shown in Figure 4. The graph clearly indicates that TCP-BFA uses less buffer space and generates smoother traffic. We also observe that Reno suffers periodic losses while there are no drops for TCP-BFA.



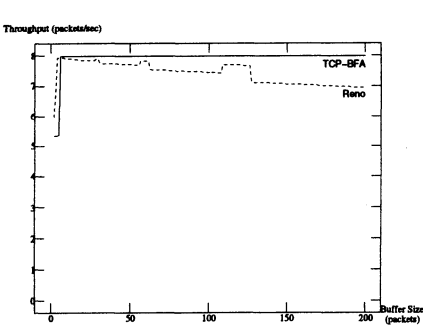
**Figure 4** Typical buffer occupancies at the bottleneck *R1* for separate simulation runs of Reno and TCP-BFA (buffer size = 20 packets).

Figures 5 and 6 show throughput and power after 400 seconds of simulation time, plotted against the bottleneck buffer size.\* Figure 5 shows that Reno's throughput falls when the buffer size increases. This happens because the maximum delay possible (the delay when the buffer is full) increases with the buffer size. A large delay causes an increase in retransmission timeout (RTO) estimates, leading to a larger recovery time when there are multiple losses. In this scenario, multiple losses occur only during the initial slow-start. These simulations were run for a constant time of 400 seconds; the decrease in throughput will be greater for connections that are more short-lived.

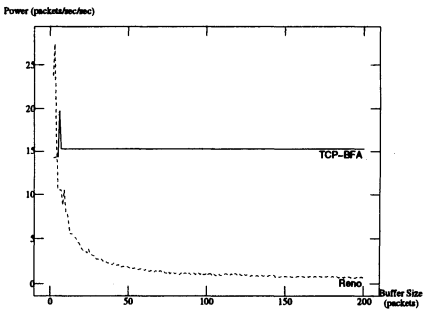
The drop in Reno's network power seen in Figure 6 is due to the compounded effects of a lower throughput and a higher delay. In contrast, for TCP-BFA, the throughput and power remain constant regardless of the buffer size. This is because TCP-BFA stops increasing its window size when the buffer starts to fill.

---

\*Note that each point on these plots represents a complete simulation run.



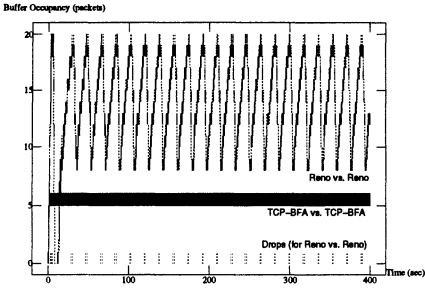
**Figure 5** Throughput against bottleneck buffer size for Reno and TCP-BFA sources running separately.



**Figure 6** Network power against bottleneck buffer size for Reno and TCP-BFA sources running separately.

4.2 Competing Sources

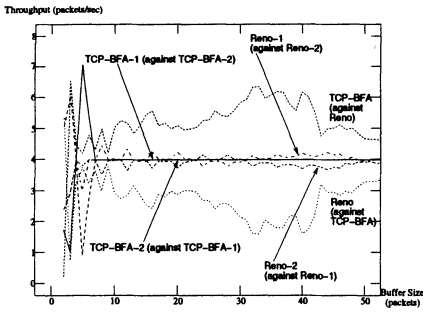
TCP-BFA sources running in competition with each other retain the benefits mentioned for a single source in Section 4.1. They avoid overuse of network buffers while maintaining high throughput and generate smooth traffic patterns. Figure 7 shows that two BFA sources (starting together) do not take any more buffer space than one source (compare with Figure 4). For Reno sources also, the average buffer occupancy stays the same but there is greater fluctuation and more frequent drops.



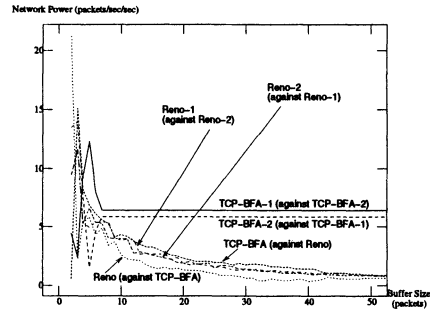
**Figure 7** Bottleneck buffer occupancy for: Reno vs. Reno and TCP-BFA vs. TCP-BFA (buffer size = 20 packets).

Figures 8 and 9 are plots of throughput and network power versus bottleneck buffer size for three different scenarios: TCP-BFA vs. TCP-BFA, TCP-BFA vs. Reno and Reno vs. Reno, with both sources starting at the same time in each case. The simulation time for each run is 1000 seconds; when the buffer size is greater than 50 packets, it takes longer to reach steady state (see Figure 10).

First we note that after a certain buffer size all scenarios have similar ag-



**Figure 8** Throughput against bottleneck buffer size for TCP-BFA vs. TCP-BFA, TCP-BFA vs. Reno and Reno vs. Reno



**Figure 9** Network power against bottleneck buffer size for TCP-BFA vs. TCP-BFA, TCP-BFA vs. Reno and Reno vs. Reno.

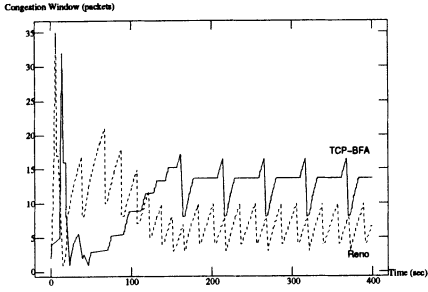
gregate throughput. This is because the buffers are never empty, hence the bottleneck link is always fully utilized. The plots show that when TCP-BFA sources compete with each other, network power is much higher than for any of the other scenarios. When TCP-BFA competes with Reno, TCP-BFA achieves higher throughput and network power. However, the network power is far less than that achieved while competing with another TCP-BFA source. This is because the Reno source will fill up the network buffers anyway, thus increasing delay. TCP-BFA will observe this same high network delay.

Figure 10 shows that TCP-BFA on average maintains a higher congestion window than Reno. This explains why TCP-BFA achieves higher throughput. The reason why Reno has smaller window sizes is that it suffers more losses. This is because, for any drop-tail buffer, packets are more likely to be dropped if they arrive in bursts. A TCP source generates back-to-back packets when its window size is increasing since multiple packets are sent for each ACK received. When the window size is constant, as in TCP-BFA's Buffer Fill Avoidance state, no back-to-back packets will be generated. Note that Reno sources will not have a disadvantage in RED buffers since they do not punish bursty sources. This conclusion has been verified by initial simulations with RED routers.

### 4.3 Fairness

Figure 8 shows that TCP-BFA sources starting together are perfectly fair,\* with exactly the same throughput. Figure 11 shows the throughput plotted against buffer size for six TCP-BFA sources starting at different times. The first source initially stops its window size increase at a higher value than the others because it was the only source in the network when it started. This

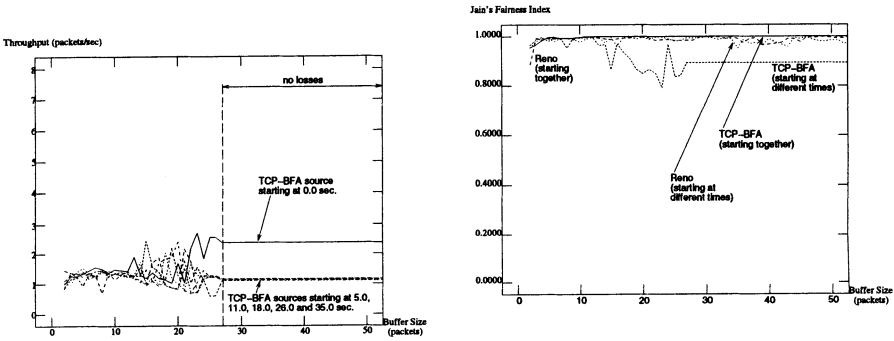
\*By fairness we mean the equal division of bandwidth between competing sources.



**Figure 10** Typical congestion window variation for Reno and TCP-BFA sources in competition (buffer size = 20 packets).

unfair situation will persist unless there are losses or changes in delay. For our simulation scenario, Figure 11 shows that there are no losses (and therefore persistent unfairness) for large buffer sizes.

Figure 12 shows plots of Jain’s (1991) Fairness Index\* ( $\phi$ ) against buffer size for six TCP-BFA or Reno sources starting either together or at different times.



**Figure 11** Throughput against buffer size for six TCP-BFA sources starting at different times.

**Figure 12** Jain’s Fairness Index plotted against bottleneck buffer size for six Reno or TCP-BFA sources, starting together or at different times.

We conducted simulations for asymmetric delay paths with two sources. For a scenario with 25% difference in propagation delays, TCP-BFA ( $\phi = 1.0000$  when starting together,  $\phi = 0.8989$  when starting at different times) was fairer than Reno ( $\phi = 0.6164$  when starting together,  $\phi = 0.6467$  when

---

\*Jain’s Fairness Index  $\phi = \frac{[\sum_{i=1}^n x_i]^2}{n \cdot \sum_{i=1}^n x_i^2}$  where  $n$  is the number of sources, and  $x_i$  is the throughput for the  $i^{th}$  source. The index is bounded between 0 and 1, with 1 indicating that all sources had the same throughput. Note that Jain’s Fairness Index is a generic metric that can be applied to any resource.

starting at different times). The drop in  $\phi$  for Reno (compared to the symmetric case) is due to the fact that sources increase their congestion windows at different rates. TCP-BFA, however, maintains  $\phi$  similar to the symmetric case because the window size is mostly constant, and the signed RTT variance is independent of the path delay.

Losses help improve fairness because they reset state and prevent persistence of an unfair situation, such as the one in Figure 11. One way to improve the fairness of TCP-BFA would be to prevent it from remaining in the Buffer Fill Avoidance state for a long time. This could be achieved by adding a timer that forces the source to switch off the BFA Flag, leading to a window size increase as in Reno, ultimately causing losses. We have chosen not to add this extra mechanism to TCP-BFA since situations without any losses or changes in delay do not exist in the real world. Internet measurements with competing sources show that TCP-BFA is fairer in real-world situations than our simulations would lead us to believe.

## 5 INTERNET MEASUREMENTS

In this section we discuss the results obtained from running TCP-BFA and Reno over the Internet. The FreeBSD 2.1.6 kernel (based on BSD 4.4) was instrumented to generate logs of various TCP state variables. The scenario is one of bulk data transfer (5MB FTP) from a 120 MHz Pentium machine running FreeBSD 2.1.6 to a DEC Alpha machine running OSF/1, across a transatlantic path of 20 hops with a 64 Kbps bottleneck link adjacent to the DEC Alpha machine. The receiver buffer size is 32KB, and the maximum segment size is 536 bytes. The round trip time for a segment of maximum size without queuing delay is approximately 320 ms. The bandwidth-delay product for this path is therefore 2.56KB, which is approximately five packets (this is the optimal window size). We note that when a (single) Reno source is running the round trip time rises to as much as 4 seconds. This implies a bandwidth-delay product of 32KB (about 60 packets), which is 12 times the optimal window size. Runs were made over a six month period at various times of day to obtain data under different network conditions.

We provide two types of plots for three different levels of congestion.\* The first is a conventional congestion window (`cwnd`) vs. time graph, which also shows the slow-start-threshold (`ssthresh`) and the receiver's advertised window. The second is a scatter-plot of RTT vs. `cwnd`; it is the real-world counterpart of the ideal plot shown in Figure 2.

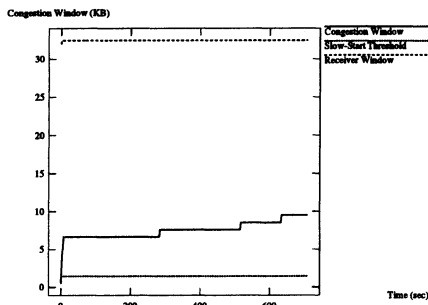
When congestion is low, the window size employed by TCP-BFA (Figure 13) is much smaller than Reno's (Figure 15). In fact, Reno's window could have

---

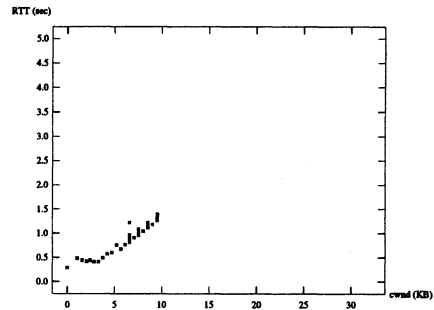
\*These three levels of congestion represent a spectrum of network conditions from 'completely unloaded' (few or no competing flows) to 'high level of background traffic' (large number of competing flows and a high level of packet loss).



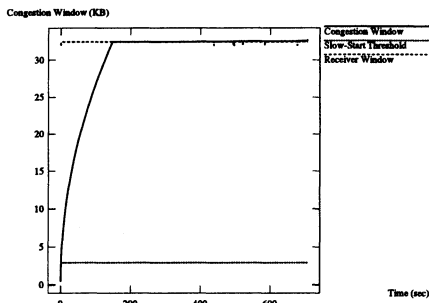
been even larger (possibly resulting in losses) if it were not for the limit imposed by the receiver's advertised window size. Despite the large difference in window size, the throughput is similar for both Reno and TCP-BFA, because the throughput of a TCP source is governed by *the rate at which the window is sliding*. The network power is much higher for TCP-BFA. The comparable throughput coupled with higher power means that TCP-BFA must have a lower delay. This is confirmed by the RTT vs. cwnd scatter-plots (Figures 14 and 16); TCP-BFA operates closer to the knee than Reno. The staircase pattern seen in Figure 13 is due to noise in RTT measurements causing the signed RTT variance to drop below  $\sigma_{off}$  momentarily.



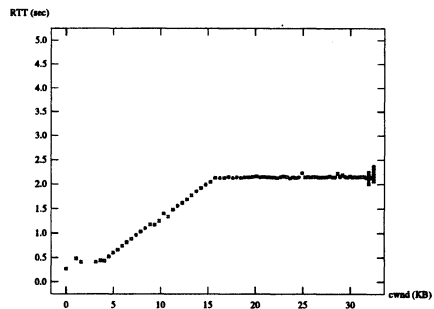
**Figure 13** Congestion Window vs. Time: TCP-BFA, low congestion. (Throughput = 56.48 Kbps, Power = 54.68 Kb/s<sup>2</sup>)



**Figure 14** RTT vs. Congestion Window: TCP-BFA, low congestion.



**Figure 15** Congestion Window vs. Time: TCP Reno, low congestion. (Throughput = 56.64 Kbps, Power = 29.02 Kb/s<sup>2</sup>)



**Figure 16** RTT vs. Congestion Window: TCP Reno, low congestion

For medium congestion (Figures 17 and 19), we note some packet losses\* and more noise in the scatter-plots (Figures 18 and 20). TCP-BFA does not achieve high power in this case because there is competition with other sources at the bottleneck. In today's Internet, we can expect most of these sources to be Reno sources. It is important to note that in this situation TCP-BFA has adapted to behave more like Reno so that it does not lose throughput to competing Reno sources.  $\alpha_{\text{srv}}$  can be increased to make TCP-BFA operate closer to the knee, though this will make it less aggressive and could cause it to yield throughput to competing Reno sources. Figures 21 and 22 show plots for TCP-BFA with  $\alpha_{\text{srv}} = \frac{3}{4}$  (instead of the usual  $\frac{1}{2}$ ). Figure 22 shows the tighter clustering of RTT vs. *cwnd* points around the knee.

TCP-BFA (Figure 23) and Reno (Figure 25) behave similarly during very high congestion. The scatter-plots (Figures 24 and 26) show that the network is not providing the sources with useful information, and the *cwnd* graphs show a large number of losses, with *cwnd* never rising over a few packets.

Unlike the simulation results described in the previous section, measurements for competing TCP-BFA sources show fair allocation of the bottleneck bandwidth. Runs were conducted with three TCP-BFA sources in competition at a time when there was little background traffic. The average value of Jain's Fairness Index for these runs was 0.9991 (the corresponding value for Reno sources was 0.9991), which is much higher than predicted by the simulations (0.89).

This results from the fact that natural losses and delay variations exist intrinsically in the real Internet. This prevents any TCP-BFA source from freezing its congestion window at a large size and monopolizing the bottleneck link. Measurements also revealed that TCP-BFA sources competing with Reno do not achieve larger throughput, but rather achieve throughputs slightly lower than Reno.

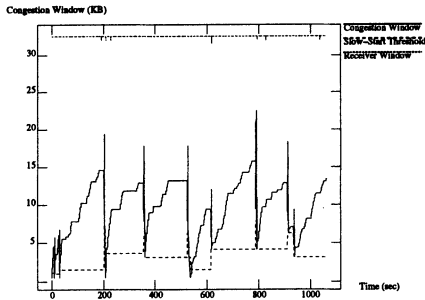
## 6 CONCLUSIONS AND FUTURE WORK

Both Reno's and TCP-BFA's congestion control mechanisms keep their congestion window size in the the region between the *knee* and *cliff* of Figure 1. The difference is that while Reno constantly drifts towards the *cliff*, TCP-BFA attempts to keep the window size as close to the *knee* as possible. Reno continues to increase the number of packets it sends into the network, even when the delay is rising. This continues until some buffer fills up and packets are lost. On the other hand, TCP-BFA will refrain from increasing its window size whenever it detects a sustained increase in delay. However, when TCP-BFA sources compete with Reno sources they are forced to operate closer to the *cliff*.

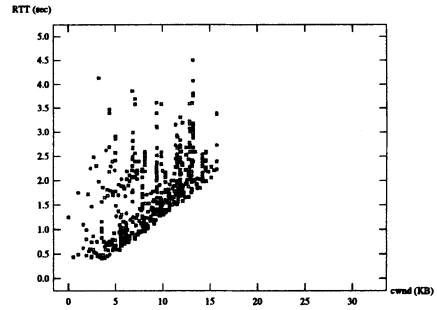
TCP-BFA retains the additive increase / multiplicative decrease mechanism

---

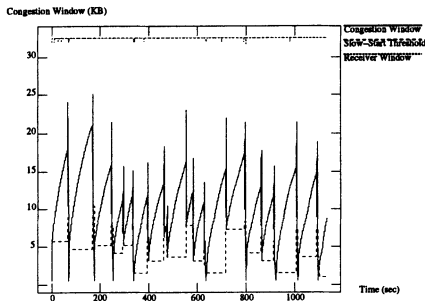
\*The spikes in these plots are due to temporary *cwnd* inflation during fast recovery.



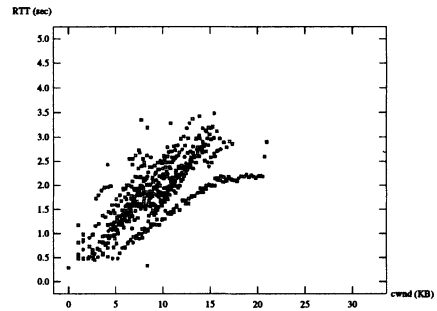
**Figure 17** Congestion Window vs. Time: TCP-BFA, medium congestion. (Throughput = 37.68 Kbps, Power = 19.84 Kb/s<sup>2</sup>)



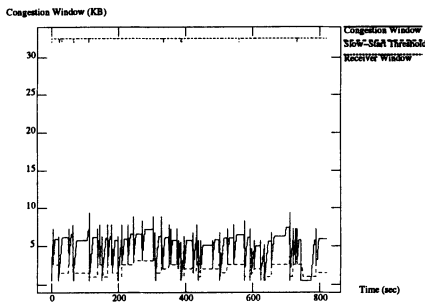
**Figure 18** RTT vs. Congestion Window: TCP-BFA, medium congestion



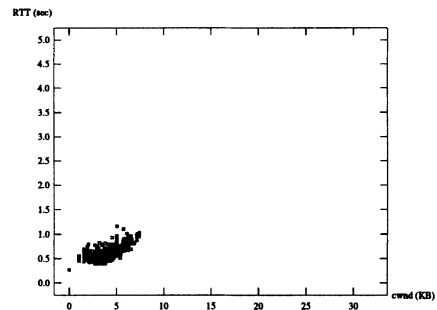
**Figure 19** Congestion Window vs. Time: TCP Reno, medium congestion. (Throughput = 35.28 Kbps, Power = 17.49 Kb/s<sup>2</sup>)



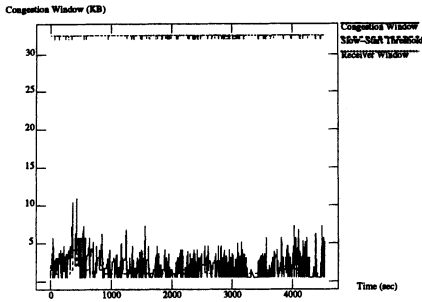
**Figure 20** RTT vs. Congestion Window: TCP Reno, medium congestion



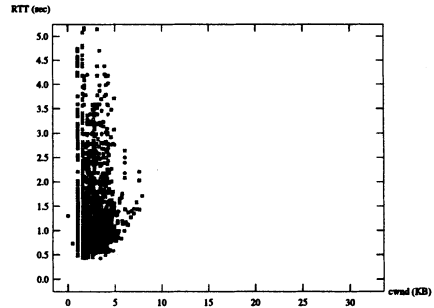
**Figure 21** Congestion Window vs. Time: TCP-BFA with  $\alpha_{srv} = \frac{3}{4}$ . (Throughput = 49.69 Kbps, Power = 70.17 Kb/s<sup>2</sup>)



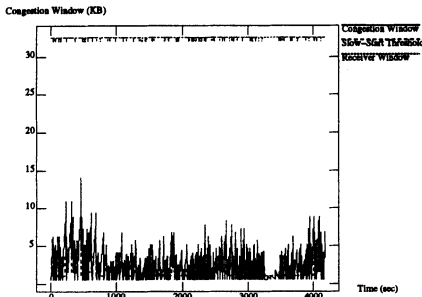
**Figure 22** RTT vs. Congestion Window: TCP-BFA with  $\alpha_{srv} = \frac{3}{4}$ .



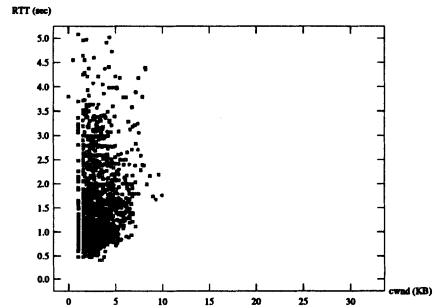
**Figure 23** Congestion Window vs. Time: TCP-BFA, high congestion. (Throughput = 8.80 Kbps, Power = 6.01 Kb/s<sup>2</sup>)



**Figure 24** RTT vs. Congestion Window: TCP-BFA, high congestion



**Figure 25** Congestion Window vs. Time: TCP Reno, high congestion. (Throughput = 9.52 Kbps, Power = 6.03 Kb/s<sup>2</sup>)



**Figure 26** RTT vs. Congestion Window: TCP Reno, high congestion

of Reno, with the addition that it sometimes halts additive increase. If TCP-BFA reaches the cliff it reacts in the same manner as Reno (multiplicative decrease). This implies that if Reno is stable, TCP-BFA is stable.

The most important benefit of using TCP-BFA is that it leads to lower network buffer occupancies – which we have demonstrated by simulations. Both simulations and Internet measurements show that TCP-BFA sources are able to achieve a considerable improvement in network power over Reno sources, except if there is high congestion, when they perform just as well. The results also demonstrate that competing TCP-BFA sources achieve higher power and experience less losses than competing Reno sources. Simulations demonstrated that TCP-BFA sources get a higher share of throughput when competing with Reno sources. However, for our specific set of measurements, TCP-BFA sources achieved slightly lower throughput. Results also demonstrated that TCP-BFA has fewer losses, confirming that TCP-BFA sources avoid the cliff. We claim that if Reno is replaced with TCP-BFA, the end-

station will get similar throughput and higher reactivity to network events due to lower delays, while the network will see lower buffer occupancies.

This work is different from Vegas in that it is simpler to implement and avoids some of Vegas' pitfalls, specifically: possible instability due to the use of finer granularity timeout values, yielding of bandwidth when competing with Reno sources, and incorrect behavior when there are route changes.\*

Future research should include the investigation of a scheme which seeks the optimal window size more aggressively, for instance by *reducing* the congestion window when the signed variance is very high. The danger for a 'nice' scheme like this is that it can lose bandwidth when operating in competition with more aggressive sources. For example, Brakmo & Peterson (1995) demonstrated that Vegas loses throughput in head-to-head transfers against Reno.

Another aspect that deserves further investigation is the dynamic estimation of the thresholds  $\sigma_{\text{off}}$  and  $\sigma_{\text{on}}$  (depending on path characteristics such as bandwidth and delay). Techniques similar to those developed in Jacobson's (1997) *pathchar* can be used to estimate these path characteristics.

The signed RTT variance that we maintain can be used to improve TCP timeout behavior. In current Reno implementations, the RTO is obtained by adding the smoothed RTT estimate to 4 times the unsigned RTT variance. However, if the signed variance is negative, RTT is decreasing and this computation can lead to unnecessarily high RTO estimates. Using a multiplicative factor of less than 4 in this situation may lead to faster packet loss detection; the smoothing of the variance should help avoid spurious timeouts.

Since at moments of high congestion the source cannot make correct inferences about the network, we believe that schemes depending on network routers to assist in congestion avoidance are necessary (e.g., Floyd's (1995) Explicit Congestion Notification, Floyd & Jacobson's (1993) Random Early Detection). Routers have a unified view of the queuing behavior over time, and can therefore make better decisions about the level of congestion compared to the endpoints which have distorted and delayed information.

Since TCP-BFA sources cause fewer losses and generate smoother traffic, it would be interesting to investigate the behavior of TCP-BFA when it interacts with real-time multimedia streams as compared to Reno sources.

We think deploying fair queuing schedulers will provide TCP-BFA with an advantage that may allow it to outperform competing Reno sources. This is because fair queuing provides a separation between different flows, hence preventing ill-behaved Reno sources from interfering with the delay of TCP-BFA sources. Demers, Keshav & Shenker (1989) emphasize that fair queuing schedulers reward sources that use more sophisticated and responsive algorithms.

---

\*TCP-BFA does not rely on a base value for RTT; instead it uses a moving average of the RTT variance.

## 7 ACKNOWLEDGMENTS

The authors would like to thank Professor Nick McKeown for his constant support and guidance without which this work would not have been possible. Much credit is due to Pablo Molinero-Fernández for initial contributions to the concepts behind Buffer Fill Avoidance. We are especially grateful to Michael Greenwald for many helpful discussions which helped shape this work. We would like to thank Nick McKeown, Craig Partridge, Yakov Rekhter and the anonymous SIGCOMM reviewers for their helpful comments on earlier drafts of the paper. We are grateful to Simon Crosby at Cambridge University, Balaji Prabhakar at MIT, and Samir Shaheen at Cairo University for providing us with access to the machines we used for Internet measurements. Finally, we would like to acknowledge the *ns* development team for their excellent network simulator.

## APPENDIX 1 DEPLOYING TCP-BFA

Deploying TCP-BFA in the Internet is a rather straightforward task. Simple changes (on the order of a few lines of code) need to be made to the TCP source; no changes are required for the sink. A patch for FreeBSD 2.1.6 network servers that can be applied to `netinet/tcp_input.c` is available at <http://klamath.stanford.edu/~aaa/tcp-bfa>. The system administrator may tune TCP-BFA's parameters ( $\sigma_{\text{off}}$ ,  $\sigma_{\text{on}}$ ,  $\alpha_{\text{srv}}$ ) to modify the behavior of the TCP-BFA source as described in Section 3.

## REFERENCES

- Ahn, J. S., Danzig, P. B., Liu, Z. & Yan, L. (1995), Evaluation of TCP Vegas: Emulation and experiment, *in* 'Proceedings of SIGCOMM '95'.
- Brakmo, L. S., O'Malley, S. W. & Peterson, L. L. (1994), TCP Vegas: New techniques for congestion detection and avoidance, *in* 'Proceedings of SIGCOMM '94', pp. 24–35.
- Brakmo, L. S. & Peterson, L. L. (1995), 'TCP Vegas: End to end congestion avoidance on a global internet', *IEEE Journal on Selected Areas in Communications* **13**(8), 1465–1480.
- Demers, A., Keshav, S. & Shenker, S. (1989), 'Analysis and simulation of a fair queueing algorithm', *IEEE/ACM Transactions on Networking* **9**(1), 1–12.
- Fendick, K. W., Mitra, D., Mitrani, I., Rodriguez, M. A., Seery, J. B. & Weiss, A. (1991), 'An approach to high-performance, high-speed data networks', *IEEE Communications Magazine* pp. 74–82.
- Floyd, S. (1995), 'TCP and explicit congestion notification', *ACM Computer Communication Review* **24**(5), 8–23.

- Floyd, S. & Jacobson, V. (1993), 'Random early detection gateways for congestion avoidance', *IEEE/ACM Transactions on Networking* 1(4), 397–413.
- Hoe, J. C. (1996), Improving the start-up behavior of a congestion control scheme for TCP, in 'Proceedings of SIGCOMM '96', pp. 270–280.
- Jacobson, V. (1988), Congestion avoidance and control, in 'Proceedings of SIGCOMM '88', pp. 314–329.
- Jacobson, V. (1990), 'Modified TCP congestion avoidance algorithm', Email to end2end-interest mailing list. Obtain via <ftp://ftp.ee.lbl.gov/email/vanj.90apr30.txt>.
- Jacobson, V. (1994), 'Problems with Arizona's Vegas', Email to end2end-tf mailing list. Obtain via <ftp://ftp.ee.lbl.gov/email/vanj.94mar14.txt>.
- Jacobson, V. (1997), 'Pathchar - a tool to infer characteristics of internet paths', ftp directory. URL <ftp://ftp.ee.lbl.gov/pathchar>.
- Jacobson, V., Braden, R. T. & Borman, D. A. (1992), 'TCP extensions for high performance', RFC 1323. (37 pages).
- Jain, R. (1989), 'A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks', *ACM Computer Communication Review* 19(5), 56–71.
- Jain, R. (1991), *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*, Wiley-Interscience.
- Jain, R. & Ramakrishnan, K. K. (1988), Congestion avoidance in computer networks with a connectionless network layer: Concepts, goals, and methodology, in 'Proceedings of SIGCOMM '88'.
- Keshav, S. (1991), A control-theoretic approach to flow control, in 'Proceedings of SIGCOMM '91', pp. 3–15.
- Kleinrock, L. (1979), Power and deterministic rules of thumb for probabilistic problems in computer communications, in 'Proceedings of the International Conference in Communications'.
- Mathis, M. & Mahdavi, J. (1996), Forward acknowledgement: Refining TCP congestion control, in 'Proceedings of SIGCOMM '96', pp. 281–291.
- Mathis, M., Mahdavi, J., Floyd, S. & Romanow, A. (1996), 'TCP selective acknowledgement options', RFC 2018.
- McCanne, S. & Floyd, S. (1997), 'Ucb/lbnl/vint network simulator - ns (version 2.1b1)', Web page. URL <http://www-mash.cs.berkeley.edu/ns/ns.html>.
- Wang, Z. & Crowcroft, J. (1991), 'A new congestion control scheme: Slow Start and Search (Tri-S)', *ACM Computer Communication Review* 21(1), 32–43.
- Wang, Z. & Crowcroft, J. (1992), 'Eliminating periodic packet losses in 4.3-Tahoe BSD TCP congestion control algorithm', *ACM Computer Communication Review* 22(2), 9–16.

## 8 BIOGRAPHY

Amr A. Awadallah is currently a graduate student at Stanford University pursuing a PhD degree in Electrical Engineering. He received the B.S. and M.S. degrees in electrical engineering from Cairo University, Egypt, in 1992 and 1995, respectively. His current research interests are in the hardware and software aspects of high performance computer network systems.

Chetan Rai is a Ph.D. candidate in computer science at Stanford University. He received the B.Tech. degree in Computer Science and Engineering from the Indian Institute of Technology Bombay, India, in 1996. His research interests range over distributed systems and computer networking, and he hopes they never narrow any further.



# Motivation of an end-to-end regulation of bandwidth in intra-networks: The ROBIN concept

*M. Frank, P. Martini*

*University of Bonn, Institute of Computer Science IV*

*Römerstraße 164, D-53117 Bonn, Germany*

*Tel.: +49-228-734550, Fax: +49-228-734571*

*E-mail: {matthew, martini}@cs.uni-bonn.de*

## **Abstract**

The variety and heterogeneity of different approaches for Quality of Service (QoS) support on different protocol layers result in the need of a solution with an end-to-end overview and regulation of data flows in a network. The ROBIN (Regulation Of Bandwidth in Intra-Networks) concept introduced in this paper is such a mechanism which controls the bandwidth used by data flows in heterogeneous networks in case of network congestion. In a project at the University of Bonn, the concept of ROBIN is designed, a prototype is implemented for laboratory measurements and a performance analysis by simulation is carried out. The paper presents an overview of the concept and reports on the state of practical work on ROBIN.

## **Keywords**

**Bandwidth regulation, congestion control, QoS support, relative quality of service, rate-based flow control, transport layer extension.**

## 1 INTRODUCTION

“The race to run time-sensitive traffic over packet-based LANs has spawned a host of approaches ... but which will emerge as the technology of choice is anybody’s guess” (Roberts, 1996). This quotation from the magazine “Data Communications International” both mentions the highly relevant issue of transmission of real-time traffic over packet-based LANs and describes the state of the art on finding an effective solution for this problem. Although this statement is already more than one year old, it has not lost a bit of relevance.

The situation has been caused by an evolutionary development of global and local networking in recent years. The technical capabilities of networks are improving rapidly, in particular the available bandwidth is rising especially in local networks (e.g. Fast Ethernet is available and becoming cheaper, Gigabit Ethernet is on its way to the desktop). Additionally, the requirements of applications change and a variety of new applications appeared on the scene bringing requirements that were not considered when classical communication protocols were designed.

Today, there are several new real-time applications that have a demand for a minimum guaranteed bandwidth (e.g. real-time transmission of audio- and video-data) and additionally may have a low upper delay bound (e.g. in a live video conference). Nevertheless, classical communication forms (like filetransfer ftp, Hypertext Transport Protocol http, News, Electronic Mail, Network File System NFS, remote login, etc.) are still alive and growing in terms of data volume. This results in a situation of competition of data flows of different applications with different requirements in a network with limited resources.

In many places, researchers and developers of networking hardware and software work on this issue. The suggestions vary from new approaches in network technologies, in particular strategies in medium access, to improvements, design or redesign of upper layer communication protocols. The efficient support of QoS (Quality of Service) properties in a heterogeneous network requires an excellent cooperation or matching of concepts, which is a very crucial point with the variety of existing or proposed approaches addressing several protocol layers.

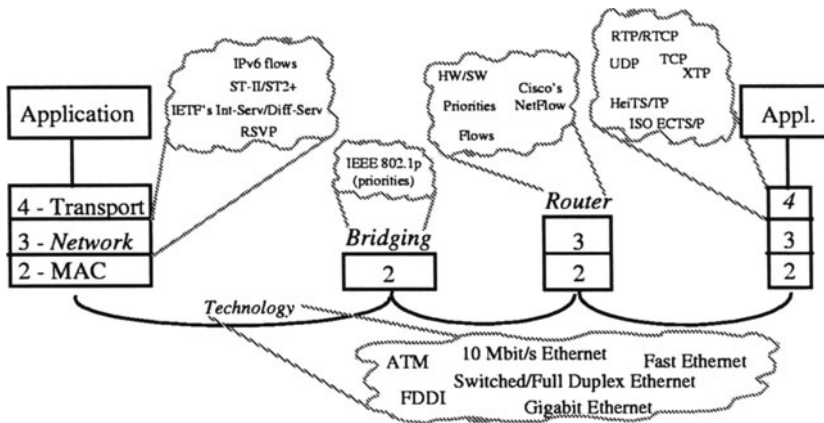
This paper introduces an extension of the transport layer (ROBIN - Regulation Of Bandwidth in Inter-Networks), which improves the effectiveness of QoS support based on the end-to-end knowledge about application data flows and a bandwidth regulation at the sending side. ROBIN has been designed for campus or intra-networks like a typical autonomous system e.g. at an university, research institute or larger company. The goal of ROBIN is to establish and maintain an (almost) optimum overall provision of the available network bandwidth to the application data flows which at the same time takes care of individual requirements of different applications. The activities concerning the ROBIN concept, realisation and evaluation are covered by a research project at the University of Bonn.

The paper is organised as follows: Section 2 presents a short review of existing and future approaches to QoS support and leads to a motivation for an end-to-end regulation of bandwidth to support different application requirements in a heterogeneous networking environment. Section 3 introduces our end-to-end

ROBIN approach with its characteristics and functionality. Section 4 presents the state of our work on ROBIN, both describing the activities on a prototype implementation and a performance analysis by simulation of ROBIN. Finally, section 5 concludes the paper and gives an outline on future work related to ROBIN.

## 2 MOTIVATION FOR END-TO-END REGULATION

As mentioned above, there are different suggestions from researchers and developers working on the issue of integrated QoS support. Figure 1 shows a selection of different classical and new QoS approaches or protocols to different layers of the OSI reference model.



**Figure 1** Overview of mechanisms/protocols with or without QoS support.

In both the transport-oriented layers of the OSI-RM (transport and network layer) and the data link layer (in particular the medium access) there are several approaches to support communication services with specific QoS requirements. In this paper we cannot discuss all of these in detail. Instead we emphasise important characteristics of the approaches and the variety of these. As important representatives of networking technologies we point at ATM (Asynchronous Transfer Mode) and Ethernet. With its connection-oriented concept ATM is well-suited for the support and guarantee of different QoS properties for different communication flows. In contrast to this, Ethernet in its basic form does not allow for any service guarantees. The development of Ethernet from classical 10 Mbit/s to 100 Mbit/s Fast Ethernet, Switched Ethernet and Full-Duplex Ethernet up to future Gigabit Ethernet includes possibilities for future QoS support. However, there is still much work to be done in this direction.

The desire to support service integration in the Internet has strongly influenced the Internet Engineering Task Force IETF in the design of the new version of the

Internet Protocol IPv6. The real use of new functions is still vague and criticism on IPv6 arises (Hutchison, 1997: "... the semantics of communication offered by IPv6 is still the well-known best-effort ..." and "...we may suspect that the support of QoS has been deliberately evicted of IPv6's design"). Additional protocols like the Resource ReServation Protocol RSVP (cf. Braden, 1997) or ST-II (cf. Delgrossi, 1996) are required to make guaranteed QoS a reality in the Internet. An important step towards RSVP is the support of resource reservation within the network connecting elements (like bridges or router) and within the networks themselves. A comprehensive overview of reservation protocols for multimedia communication may be found in (Delgrossi, 1996).

At the end of last year a new working group (Differential Services for the Internet – Diff-Serv) of the Internet Engineering Task Force (IETF) split from the Integrated Services (Int-Serv, cf. Baker, 1996) working group. The differential or differentiated services (both terms are in use) intend to introduce a service discrimination in the Internet without the need for per-flow state and signalling at every hop (cf. Nichols, 1998) and without requiring admission control for flows using this service. Several proposals for service categorisation and frameworks for deployment of these services are under discussion (e.g. Kalyanaraman (1998) or Nichols, 1997). Without having direct relation to activities of Int-Serv and Diff-Serv working groups, Braden (1998) indicates some recommendations on active queue management and scheduling algorithms for congestion avoidance of best-effort traffic in the Internet. However, no consensus solution is existing to control congestion caused by unresponsive or not sufficiently responsive flows (not adapting to indications of congestion like TCP does). Also Floyd (1997) addresses these problems and also emphasises the negative impact of non-controlled best-effort traffic on the Internet in terms of "extreme unfairness ... to the potential for congestion collapse". They suggest per-flow scheduling and router support for end-to-end congestion control.

Independent of the underlying transport and network layers, the Real-time Transport Protocol RTP (cf. Schulzrinne, 1997) has been designed by the IETF to support real-time communication. RTP and RTCP (RTP Control Protocol) provide identification of payload type, sequence numbers, timestamps and report feedback about the network status allowing the application to do the synchronisation and media scaling. However, there is no negotiation of QoS parameters prior to the data transmission and in case of congestion no active actions of RTP are performed. Instead, the application is expected to take care of these problems. Applications typically run RTP on top of UDP, which offers no functions of QoS support that RTP could use. Furthermore, with RTP/RTCP it is not possible to estimate the impact of other (possibly malicious) protocols operating in the same network.

The title "Internet Protocol Quality of Service Problem Statement" of the Internet Draft (Bradner, 1997) already points at the problem about QoS in IP: Bradner (1997) lists a number of essential but extensive requirements for effective support of QoS in the Internet Protocol. Some of the approaches are questionable already because of intrinsic problems. The situation becomes much worse in larger campus

or intra-networks consisting of up to several hundreds of hosts, as it is characteristic for universities, research institutes or larger companies. Typically, in such environments we find heterogeneous interworking of different network technologies and topologies, connected either by hosts operating as software routers or by hardware components (bridges, routers, switches and hubs). We expect that this situation will remain typical for such networks since large local and campus networks usually have heterogeneous structures resulting from incremental growth and replacements in specific parts of the network. A complete move to a homogeneous technology would probably bear too much cost to afford.

The provision of QoS properties in such a heterogeneous network requires cooperation or matching of concepts, which is a very crucial point with the variety of existing or proposed approaches covering several protocol layers. These approaches are far from being intuitively combinable. The argument of heterogeneity, in particular on layers 2 and 3 of the OSI RM, leads to the conclusion, that the control mechanism to support QoS should be moved to or added above the network layer with an end-to-end knowledge of different transport protocols, their active data flows and the current network status. The traditional end-to-end view of transport protocols is not enough for this, because the congestion reaction scheme (e.g. as with TCP) does not know details (e.g. bandwidth requirements or actual bandwidth use) of other communication flows, neither of the same protocol nor of other transport protocols.

The idea to make use of global or end-to-end knowledge on active data flows and network state for an integration of QoS support has already been considered at the University of Virginia: TReg - Transport Regulation (cf. Liebeherr, 1995) enhances applications on top of the TCP/UDP protocols to control the application data flows. With the TReg concept of Liebeherr (1995), all application processes using the service of TCP or UDP have to be enhanced by adding TReg components (so-called "TReg-Stub" and "TReg-Daemon"). These components have a global view on all active TReg flows in the network. They know about the utilisation of network resources and are directly able to control the source of each flow within the application process before having access to the transport protocols. However, the TReg approach suffers from the possibly malicious interference of other applications and transport protocols that do not implement the TReg mechanism. A closely related work (cf. Akyildiz, 1996) implements the bandwidth regulation within the network layer, which obviously implies an adaptation of the network protocols including the inter networking units (possibly difficult to realise in bridges or hardware routers).

The Lancaster QoS-Architecture (QoS-A, cf. Campbell, 1994) defines an "integrated and coherent framework that incorporates QoS interfaces, management and mechanisms across all architectural layers" (Campbell, 1994). QoS-A has a focus on an integration of QoS properties in both end-systems and the network with a support of quantitative end-to-end QoS. The ROBIN approach with its end-to-end knowledge on active data flows very well matches the architectural model of QoS-A to represent one possible option of QoS support on the transport layer. This way, the multi-layered approach of QoS-A may help ROBIN in achieving its

goals. However, the ROBIN approach is free from depending on a quantitative specification of QoS properties prior to the transmission and can already work without support from the network and its hardware/software elements.

The working group ISSLL (Integrated Services over Specific Link Layers) of the IETF presently discusses a framework for providing integrated services on OSI-RM layer 2 over shared and switched IEEE 802 LAN network technologies (cf. Ghanwani, 1998). The goal is to control the resources of network segments and bridges/switches at the data-link layer with a "Subnet Bandwidth Manager" (SBM, cf. Yavatkar, 1998) to support integrated services and quality of service provision. The approach of SBM is designed to work without dependencies on particular upper layer protocols like RSVP or IP. Within the framework and architecture of SBM on OSI layer 2 and our ROBIN concept on OSI layer 4 there are several related aspects: Ghanwani (1998) introduces several possibilities of implementing the bandwidth manager (centralised, distributed or semi-distributed), which has to manage the resources of the entire subnet. Aspects like soft state with periodical refresh and good scalability are essential requirements. As with ROBIN, the flow separation and scheduling is an important issue. The documents of the ISSLL group on SBM are Internet Drafts documenting the state of work in progress and there is no information on practical experience available yet. However, the progress of this work has to be studied carefully to make use of solutions on common issues with our ROBIN work.

The next section presents our approach to extend the transport layer to support QoS properties by bandwidth regulation of different transport protocols and their data flows.

### 3 THE END-TO-END CONCEPT OF ROBIN

ROBIN is operating as an extension of the protocol stack between transport and network layer protocols. ROBIN regulates the bandwidth by a rate control at the source of individual data flows. Due to the end-to-end operation of ROBIN, no active cooperation of network protocols and inter-mediate network nodes is required. The ROBIN mechanism is not connection-oriented in the sense that a negotiation of QoS parameters for each flow must be done prior to the data transmission. Instead, the goal of ROBIN is to establish and maintain a nearly optimum overall utilisation of the available network bandwidth by the application data flows which at the same time takes care of individual requirements of different applications.

Basic ideas on the improvement of QoS support with ROBIN are as follows:

- In a campus network that does not support any QoS functionalities at network or lower layer protocols, ROBIN is able to considerably improve this situation by providing a relative QoS based on the different requirements of different applications.
- ROBIN is able to control the sending bandwidth of flows unresponsive to congestion and thus is able to improve the fairness of network bandwidth distribution between all active flows.

- ROBIN is able to make use of or cooperate with different QoS approaches on layers 2 and 3 with a mutual benefit and increase of effectiveness: E.g. reservation protocols or priority schemes may support ROBIN in providing its end-to-end QoS and in turn ROBIN will extend the efficiency of other schemes with its end-to-end knowledge of the data flows in a campus network.

### 3.1 Basic concept of ROBIN bandwidth regulation

The basic concept of ROBIN bandwidth regulation has been published in (Frank, 1997). The feedback and experiences gained within the first steps of implementing the ROBIN concept as a prototype and for simulation study have already led to some improvements which are included in the following description of the ROBIN concept.

ROBIN is operating as an extension of the protocol stack between transport and network layer protocols in the operating system of each station of the network. Each instance of ROBIN has knowledge about the complete network topology, the capacity of network elements and the individual bandwidth requirements of "own" flows leaving this station. This "own" information is distributed to the peer ROBIN instances within a frequent state exchange message. Within this, all ROBIN instances approximately have the same knowledge of the complete network state and active data flows. In case of overload of particular elements in the network ROBIN distributes the bottleneck bandwidth between the competing flows by regulating the sending bandwidth in an appropriate way. The overhead for state exchange and additional calculation limits the area of ROBIN regulation to campus or intra-networks like a typical autonomous system e.g. at a university.

With the calculation of the sending bandwidth of individual flows, ROBIN achieves a nearly optimum utilisation of the bandwidth available in the network for all active data flows across this network. All existing local data flows of transport protocols like UDP, TCP and special purpose protocols are considered. Therefore, ROBIN is able to include congestion unresponsive flows (like UDP or other transport protocols) in its bandwidth distribution. The available bandwidth of the network (in particular parts of it being a bottleneck) is shared according to the principle of relative QoS as used in (Liebeherr, 1995): Different applications are classified according to their requirements in terms of minimum bandwidth (e.g. for video) or relative priorities (e.g. video is more important than filetransfer). The available network bandwidth for each class is equally shared by all active communication flows of this class (e.g. all active video connections). However, the reserved share of the total bandwidth differs between classes according to their priority. The definition of classes, mapping of applications to classes and the reservation of bandwidth for each class in all network components is specified in advance by the system administrator.

The main advantage of sharing the bandwidth with this principle of relative QoS is that this approach satisfies the bandwidth needs better than a best-effort network service without additional knowledge about flow requirements and thus sharing the bottleneck bandwidth evenly between the competitors, if the latter is met at all.

E.g. the unfairness problem of TCP is well-known and several approaches have been proposed to improve these drawbacks (cf. the proposal of Satyavolu (1998) or related work cited there).

Where possible, ROBIN also supports the negotiation of a certain transmission bandwidth during connection set-up. Applications that are able to specify their bandwidth demand in advance (with similar parameters as for RSVP set-up), may reserve bandwidth from ROBIN. If the connection is accepted then these flows are given higher priority in the bandwidth allocation (class 0 flows, cf. Table1).

**Table 1** Possible classification of applications for relative QoS

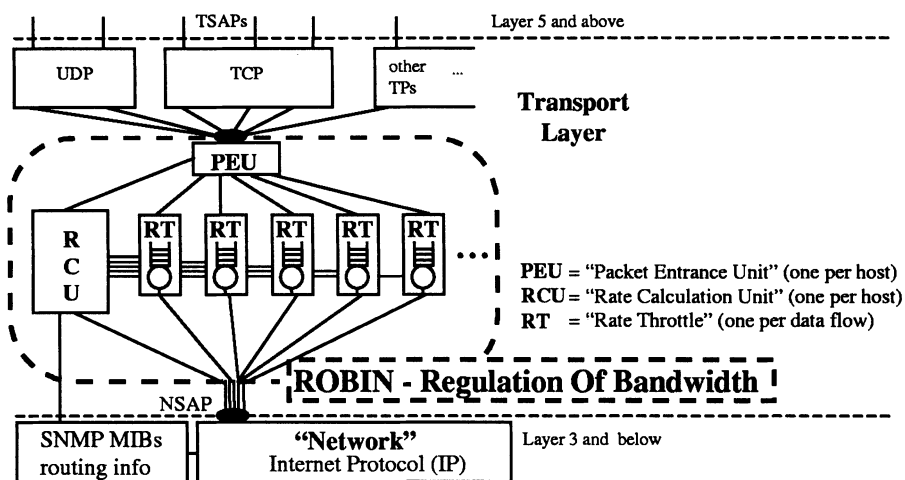
<i>Class</i>	<i>Share of bandwidth (%)</i>	<i>Applications type</i>
0	Absolute fraction of bandwidth	All applications that are able to negotiate desired bandwidth in advance
(1-4	Classes 1-4 share the remaining bandwidth)	
1	up to 50	Audio, video, multimedia streaming
2	up to 30	WWW, ftp
3	up to 15	E-mail, news, software distribution
4	up to 5	Telnet, rlogin

Table 1 shows an example of mapping of applications to classes and the reserved bandwidth share of a bottleneck in the network (here with a total of 5 classes). The reserved bandwidth shares may be selected individually for particular network segments, e.g. in network elements close to a video server the fraction of class 1 may be higher than in other parts of the network.

If 4 video flows (class 1) and 10 ftp-flows (class 2) have to share the bandwidth in a 10 Mbit/s Ethernet segment, with the principle of relative QoS each video flow gets 1,75 Mbit/s ( $70\%/4$ ) and each ftp-flow 0,3 Mbit/s ( $30\%/10$ ), where the unused reservation of classes 3 and 4 has been used by class 1. Without such a regulation by ROBIN, all flows had to share the available bandwidth equally ( $10 \text{ Mbit/s}/14 = \text{approx. } 0,7 \text{ Mbit/s}$  for each flow), which would lead to a considerable degradation of quality for the video flows.

Figure 2 depicts the components of a ROBIN instance and the information flow between these components. The functionality of each component is explained in the following.





**Figure 2** ROBIN architecture.

### *Packet entrance unit (PEU)*

Each data segment passing from transport to ROBIN is handled by the PEU and according to its data flow identification is directed to the corresponding rate throttle (RT). Flows and their application classes are identified by source and destination IP addresses, port numbers and/or transport protocol identifier. This identification of data flows within a protocol below the transport layer is an adoption of ideas from the Integrated Services Architecture ISA of the IETF (cf. Baker, 1996).

If there is no RT existing for this flow (i.e. a new transmission started), the PEU generates a new RT, passes the segment to the new RT and also notifies the RCU about the new flow. Within the IP-like access to ROBIN via the PEU we achieve a transparency of ROBIN from the transport protocol's point of view. Correspondingly, the integration of ROBIN into an existing stack of protocols in the TCP/IP family is rather simple. Furthermore, with this approach the data flows of all transport protocols are controlled by ROBIN bandwidth regulation, even those of connection-less protocols like UDP.

### *Rate throttle (RT)*

There is one rate throttle for each active data flow leaving the ROBIN instance. The main task of the RT is to measure the throughput of data sent from the application and transport protocol of the corresponding flow. If activated by the RCU, the RT limits the bandwidth available to this flow with a rate control according to the parameters provided by the RCU. Due to the automatic generation of the RT by the PEU (without any connection or flow set-up), the RT does not learn explicitly about the end of transmission for a flow. Instead, the RT is removed after a certain inactivity time-out.

The limitation of sending bandwidth within the RT is established by a rate control with parameters <rate> and <burst> (as with rate control in XTP, cf. XTP-Forum, 1995). This scheme is comparable to the classical token bucket algorithm as used for traffic shaping, where our sending RT is able to save up permission to send <burst> bytes without constraint. In case of suspended transmission, the RT may apply a blocking operation to give backpressure to the transport protocol, or may even work in non-blocking mode with queuing and scheduling in a fair queuing manner. Currently, different realisations of RT modes are under discussion and will be evaluated in further studies.

The actions of the rate throttle to regulate the bandwidth by delaying or in the worst case even discarding packets may cause an interference with mechanisms of transport protocols. These will be studied in detail to avoid a reduction of the efficiency of those protocols. Nevertheless, this interference may be used for a passive cooperation of ROBIN with transport mechanisms: e.g. TCP's congestion control is sensitive to round-trip delay and packet loss and may adapt its sending bandwidth to the available bandwidth coming from ROBIN regulation. Well-known approaches for adaptation of queue management like Random Early Detection (RED, cf. Braden, 1998 or Floyd, 1997) or new proposals for TCP support with per-flow queuing (cf. Suter, 1998) to improve performance of reactive end-to-end congestion control mechanisms will be considered for buffering in the rate throttles of ROBIN as well.

In addition, an active cooperation with transport protocols is useful, when the protocols are able to use information from ROBIN: The calculated sending rate from ROBIN may be used by the transport protocol to scale down its data flow accordingly or to pass this information to the application to react (e.g. similar to what RTP/RTCP does by its QoS monitoring).

### *Rate calculation unit (RCU)*

The RCU is the most important component of ROBIN: It maintains a data structure representing the topology of the network, the bandwidth capacity of all network elements operating in the network, and the classification of application flows and per class bandwidth reservation for realising the relative QoS. Each RCU knows the details (application class, presently consumed bandwidth, bottleneck location) of the "own" flows leaving the host where it resides. It distributes this information to all peer RCUs in frequent intervals of time by a multicast or broadcast state message. By this, all RCUs have almost the same global knowledge of the number and properties of all active flows in the network and are able to determine bottlenecks and to limit the bandwidth of flows involved to avoid congestion.

The network topology is automatically detected by a tool working with SNMP (Simple Network Management Protocol) and internally represented as a graph. The routes of flows in the network are internally stored after using information from routing protocols. We assume a static routing in IP in use within the network under control of ROBIN. This assumption is no limitation, because static routing is very often used in campus like networks.

The algorithm to calculate the sending rate of the RTs for each flow works in two phases: In the first phase, the algorithm follows all routes of own flows from the source to the destination within the internal network graph. For each network element with an overload caused by own and foreign flows passing through, the bandwidth shares of all flows are calculated according to the principle of relative QoS. It also considers reserved bandwidth of empty classes, which is immediately distributed to classes with active flows. In the same step, the bottleneck of each flow is determined when following the routes of the flows and the sending rate is adapted to the bottleneck bandwidth. After this phase, there may be a situation in single nodes with several flows, where some of these flows have their bottleneck at this node and flows having a different bottleneck node. Thus, the latter do not use their calculated share completely and bandwidth is left unused. Additionally, some flows may not completely use their calculated share of the relative QoS principle without having a bottleneck at all. In both cases, the unused amount of bandwidth should be re-used by bottlenecked flows in these nodes.

Phase 2 tries to re-use possibly unused bandwidth in all bottleneck nodes of own flows. Unused bandwidth of particular flows is first distributed to bottlenecked flows of the same class and second to flows of other classes (with an order of reuse or priority from classes 1,2, ... down). With all flows belonging to the same class, the ROBIN regulation very well meets the requirements of MaxMin Fairness (cf. Simcoe, 1994). The bottleneck rate for a flow is achieved, unused bandwidth shares are divided equally among bottlenecked flows and after a change in network state, the sending rates are adapted to the new situation accordingly. But also the classification of flows according to relative QoS with different amounts of bandwidth reserved for each class introduces no unfairness and does not violate the MaxMin fairness.

### 3.2 Critical issues on ROBIN operation

#### *Data flows to/from outside the intranet*

The area of ROBIN regulation is limited to a campus network with limited geographical extension. Obviously, there are connections or data flows running to and from the network located outside the ROBIN intranet. The flows leaving the intranet are under control of ROBIN on their route from the source to the border of the network (e.g. the router or gateway to the WAN) and the sending rate may be limited to adapt to overload within the campus network.

At the sending station of flows entering the campus network no ROBIN instance is participating in the bandwidth regulation to control these flows. In most cases the connection of a campus network to a WAN (like the Internet) is of a rather small bandwidth compared to the bandwidth within the local network. Thus, a dedicated ROBIN instance at the "entrance" to the campus network could passively quantify the incoming external traffic and consider this amount within the calculations on the route from the entrance router/gateway to the destination subnetwork and host. Additionally (or in particular with a faster connection to the outside world), an active component could perform a traffic shaping of incoming

data flows. Commercial products are available for this purpose in the IP-protocol family (cf. "Packetshaper", Bruno, 1996), which are able to control incoming flows of particular applications, destination subnetworks or even individual hosts.

### *Scalability of ROBIN*

The first version of the ROBIN concept was based on the assumption that each ROBIN instance distributes its information about own flows frequently to all other ROBIN instances, which register the data and have the knowledge of the complete network present all the time. Although ROBIN usage is limited to a local area, the overhead for distribution of state information and the calculations based on the complete knowledge may already limit the scalability to the order of typical intra-networks.

To improve the scalability of the ROBIN approach, we follow different solutions which may be combined. Up to now, the state exchange and re-calculation of sending bandwidth was done after fixed time intervals. An estimation of communication overhead with 2000 active data flows showed for the worst case (large state exchange packets, large overhead for lower layer protocols as IPv6 + FDDI, state of each flow sent in a separate packet) that an exchange interval of one second results in an additional network load of 1 Mbit/s. In classical 10 Mbit/s Ethernet segments, this is an overhead of 10% for realising the ROBIN concept. To ease this situation, a dynamic determination of points in time for distribution of state information and for re-calculation reduces the overhead. A compromise between "as little as possible" and "as much as necessary" has to be found: Only after relevant changes in the system (such as additional flows, flows leaving, change of parameters within pre-defined tolerance), rates will be re-calculated and a state packet will be sent.

An additional approach would be a hierarchy according to the principle "divide and conquer": The total network is divided into several ROBIN-islands with only local knowledge about island-internal topology and internal flows. The state information is exchanged on two levels: Firstly, within an island for all internal flows, secondly between islands only regarding the flows between islands, where one island does not know all details of other islands. This approach reduces overhead for state exchange and complexity of calculations and additionally allows for a connection of several remote ROBIN islands (e.g. two remote company network sites connected by a fast direct link).

These extensions of the basic concept are presently under preparation and will be covered by detailed studies of performance and scalability within a comparison of different versions of our concept.

## 4 IMPLEMENTATION OF ROBIN

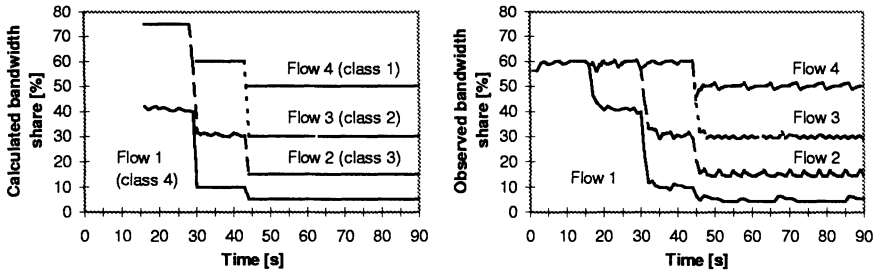
Within the scope of the project the concept of ROBIN is designed, specified and implemented as a prototype. The implementation aims at verifying the functionality of ROBIN in a local network environment. Due to the fact that performance measurements giving sound results in a prototype system are only

possible with some limitations, a performance analysis with modelling and simulation of the ROBIN system will also be done. This simulation study will evaluate the impact of certain parameters on the ROBIN functionality and will be able to analyse the ROBIN performance in larger scenarios than possible with the prototype implementation.

The ROBIN prototype system has been specified in SDL (Specification and Description Language) and automatically implemented with our proprietary tool SDL2C (translating SDL specifications to C-code). The prototype is running on a UNIX platform. The modelling and program implementation for the simulation study is done with the help of our proprietary simulation tool OOSIM (Object-Oriented SIMulation Library), a library to support discrete, asynchronous, event-driven simulation. A detailed description of the simulation model for ROBIN studies has been presented in (Frank, 1998). For the experimentation with our concept of ROBIN we follow an integrated approach of specification and implementation for the prototype realisation and the simulation study of ROBIN. The interfaces to the network and transport protocols, the PEU and the RTs have specific implementations for the prototype and for simulation. The SDL specification with processes, internal signals and interface functionalities had to be transformed into the simulation model of PEU and RTs and to corresponding simulation events. The RCU, the data structures for network topology and flow characterisation and in particular the algorithm to calculate the bandwidth shares for each flow have been integrated directly in C and thus can be used for prototype implementation and object oriented simulation without further adaptation. This allows for a close mapping between prototype implementation and simulation activities and a simple exchange of adaptations to the concept between the two of these.

The objective of the prototype measurement presented here was to demonstrate the principle of relative QoS and the effectiveness of ROBIN to achieve this principle. Each flow belongs to a different class (1 .. 4) and generates a load of 60% of the bottleneck bandwidth (in an Ethernet network segment). Thus, with 2 or more flows active, there is a state of overload. Without any control mechanisms in the transport protocol (like with UDP) the flows are unresponsive to indications of congestion. The flows are starting at regular intervals of 15s. The reserved bandwidth for each class is equal to the example classification given in Table 1. Figure 3 shows the bandwidth shares of the four flows resulting from measurements on the prototype. The left graph depicts the shares of sending bandwidth calculated by ROBIN and the right graph shows the actual throughput of the flows (after a possible regulation), both displayed over the time.

With only flow 1 active (0 - 15 s), there is no rate limitation of ROBIN and there are no plots in the left graph yet. From 15 - 30 s, flows 1 and 2 are active which results in overload. The initial share is calculated from the bandwidth reserved for each class with active flows: Classes 3 and 4 have active flows and have a reservation of  $15\% + 5\% = 20\%$  of the total bandwidth. The reservations of classes 1 and 2 are unused and are immediately distributed to classes 3 and 4: Class 3

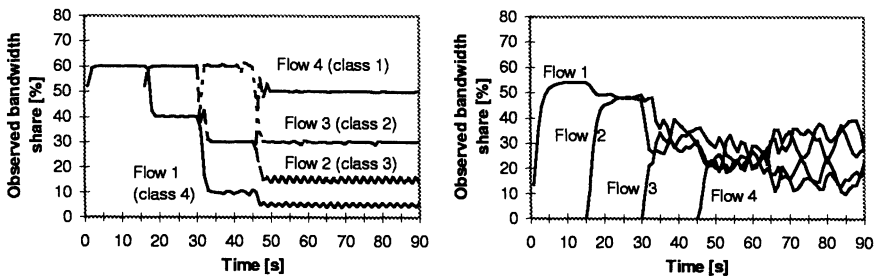


**Figure 3** Prototyping measurement: (a) calculated, (b) measured bandwidth shares.

receives  $15/20 = 75\%$  and class 4 receives  $5/20 = 25\%$  of the total bandwidth. The actual demand of flow 2 (of class 3) is only 60%, thus the 15% of unused bandwidth are reused by flow 1 (resulting in a regulation of  $25\% + 15\% = 40\%$ ). Nevertheless, a rate limitation of flow 2 with a value of 75% is activated for safety reasons. From 30 - 45 s, the shares of the three active flows are  $30/50 = 60\%$ ,  $15/50 = 30\%$  and  $5/50 = 10\%$  for classes 2, 3 and 4 respectively. After 45 s all flows are active and the calculated shares equal the class reservations as indicated in Table 1.

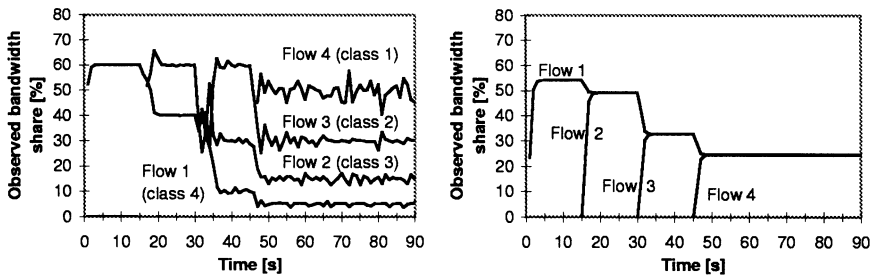
The relative QoS principle with ROBIN bandwidth regulation is clearly visible: The bottleneck bandwidth is purposely shared unequally between flows from different application classes. Also the dynamics of the rate adaptation is illustrated with starting flows: The plots of calculated rates achieve the new bandwidth share value instantaneously after a change in the network. In contrast to this, the measured bandwidth share reflects the impact of the ageing function used to smooth the measured throughput.

Due to performance limitations with the prototype implementation, it is not possible to make a fair comparison of performance in a scenario with and without ROBIN operation. Thus, this comparison has been performed by the simulation study: Figure 4 shows the measured bandwidth for the previous scenario with and without ROBIN bandwidth regulation.



**Figure 4** Simulation result: Bandwidth shares (a) with, (b) without ROBIN.

Figure 4 (a) is almost identical to Figure 3 (b), which demonstrates the correct functionality of ROBIN in both the prototype measurement and the simulation study. Without ROBIN operation, the measured throughput on the receiving sides oscillates rather heavily and there is even an unfairness between the four flows visible after about 65s. Again it has to be noted that we used no transport layer mechanisms for congestion control to exclude an interference with ROBIN operation in our first studies. Thus, the situation without ROBIN operation is caused by an overload of the network as it is the case with several unresponsive UDP data flows sending to the same network segment.



**Figure 5** Bandwidth shares in (a) Ethernet with ROBIN capacity 90 Mbit/s), (b) FDDI without ROBIN regulation.

The capacity of the FastEthernet segment for ROBIN bandwidth regulation was selected to 80 Mbit/s (in Figure 4 (a)), which is shared amongst all active flows. A higher capacity of 90 Mbit/s already caused frequent oscillations of the measured throughput due to collisions on the medium (cf. Figure 5 (a)). Thus, the capacity of particular network elements as an input parameter for ROBIN as target value for maximum network utilisation has an important impact on the performance with the bandwidth regulation. Furthermore, the achievable utilisation depends on the type of the network, as it was seen for an Ethernet type where the collisions on the medium restricted the maximum utilisation.

This is completely different for a network with a deterministic medium access like FDDI. Figure 5 (b) shows the FDDI results on measured throughput without ROBIN bandwidth regulation (as Figure 4 (b) did for an Ethernet network). At all points in time, the available network bandwidth is equally shared between all active flows. The high aggregate throughput of 96,5% is achieved by the implicit flow control of the FDDI medium access. Thus, with a bandwidth regulation with ROBIN, a target network utilisation of more than 90% is achievable without degradation of the overall throughput. However, when selecting the values for the capacity of particular network elements and network segments under control of ROBIN, the delays with a high target network utilisation and additional delays coming from ROBIN's limitation of the sending rate have to be considered carefully. Our studies will be continued soon to investigate these aspects.

## 5 CONCLUSIONS AND FURTHER WORK

Already our short review on existing and future approaches to QoS lead to the conclusion, that the effective support of QoS in a typically heterogeneous network requires close cooperation or matching of concepts. With the variety of existing or proposed approaches, this is a very crucial point and presently hampers an effective support of QoS in such networks. Our paper introduced the new ROBIN concept of end-to-end regulation of bandwidth in local or campus area networks to control the sending bandwidth of application data flows with respect to a relative quality of service principle. In addition, we presented the state of our practical work on the implementation and performance evaluation of the ROBIN concept.

Altogether, the overall objectives of first prototype measurements and simulation runs have been met: The main goal of providing a relative QoS concept by the ROBIN mechanism is achieved and the RCU is determining a bandwidth bottleneck and the corresponding shares correctly. The benefit of the ROBIN approach and its relative QoS was underlined by a comparison to the operation without ROBIN in different network types without further adaptive flow or congestion control (as with connection-less UDP flows or other unresponsive protocols). The approach of ROBIN has the same goals as Braden (1998) suggesting active queue management and per-flow scheduling in routers for preventing unfairness and congestion collapse in the presence of unresponsive flows. In contrast to these approaches using mechanisms in the network, ROBIN indirectly meets the recommendations of Braden (1998) by adding some functionality to the sending source hosts. ROBIN establishes its principle of bandwidth scheduling on an end-to-end basis with a per-flow regulation at the sending hosts.

Further measurements with the prototype implementation will be carried out, probably leading to new iterations of concept enhancement, implementation and experimentation. These measurements in particular will focus on the evaluation of different RT modes (as mentioned above) and on the effectiveness of re-use of bandwidth of empty classes or of unused bandwidth of single flows with bottlenecks in different networks.

In parallel, the simulation studies of ROBIN performance will be extended: The performance analysis will be extended to larger and more complex scenarios as to be studied with prototype measurements. These simulation studies will in detail evaluate the impact of certain ROBIN parameters on performance and overhead and intend to investigate bounds on scalability of the ROBIN approach. The inclusion of typical LAN traffic with many short-lived flows into our experiments is an important step and may have considerable impact on ROBIN performance and efficiency of its bandwidth regulation. These aspects have to be evaluated carefully as well. However, a dynamic distinction and separation of packets belonging to long-lived and short-lived flows with a regulation of the aggregate bandwidth of short-lived flows will allow ROBIN to achieve its goals under these circumstances. The continuation of our studies probably will give hints for further improvements of the concept.



## 6 REFERENCES

- Akyildiz, I., Liebeherr, J., Sarkar, D. (1996) Bandwidth Regulation of Real-Time Traffic Classes in Internetworks. *Computer Networks and ISDN Systems*, Vol. 28, No. 6.
- Baker, F. (1996) Real-Time Services for Router Nets. *Data Communications International*, Special Issue May.
- Braden, B., Zhang, L., Berson, S., Herzog, S., Jamin, S. (1997) Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification. *RFC 2205*.
- Braden, B., Clark, D., Crowcroft, J. et al (1998) Recommendations on Queue Management and Congestion Avoidance in the Internet. *RFC 2309*.
- Bradner, S. (1997) Internet Protocol Quality of Service Problem Statement. *Internet Draft <draft-bradner-qos-problem-00.txt>*, IETF.
- Bruno, L. (1996). The Internet Gets a Traffic Cop. *Data Communications International*, Vol. 25, No. 17.
- Campbell, A., Coulson, G., Hutchison, D. (1994) A Quality of Service Architecture. *ACM Computer Communications Review*, Vol. 24, No. 2.
- Delgrossi, L. (1996) Design of Reservation Protocols for Multimedia Communication. Kluwer Academic Publishers, 1996
- Floyd, S., Fall, K. (1997) Router Mechanisms to Support End-to-End Congestion Control. *Technical Report LBNL Network Research Group*.
- Frank, M., Martini, P. (1997) Practical Experiences with a Transport Layer Extension for End-to-End Bandwidth Regulation. *22nd Annual Conference on Computer Networks*, LCN '97, Minneapolis.
- Frank, M., Martini, P. (1998) Performance Analysis of an End-to-End Bandwidth Regulation Scheme. *Sixth International Symposium of Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, MASCOTS '98, Montreal.
- Ghanwani, A., Srinivasan, V., Smith, A., Seaman, M. (1998) A Framework for Providing Integrated Services Over Shared and Switched IEEE 802 LAN Technologies. *Internet Draft <draft-ietf-issll-is802-framework-04.txt>*, IETF.
- Hutchison, D., El-Marakby, R., Mathy, L. (1997) A Critique of Modern Internet Protocols: The Issue of Support for Multimedia. *Proceedings of 2nd European Conference on Multimedia Applications, Services and Techniques - ECMAST '97*, Milan.
- Kalyanaraman, S., Harrison, D., Arora, S., Wanglee, K., Guarriello, G. (1998) A one-bit feedback enhanced differentiated services architecture. *Internet Draft <draft-shivkuma-ecn-diffserv-01.txt>*, IETF.
- Liebeherr, J., Tai, A. (1995) A Protocol for Relative Quality-of-Service in TCP/IP-based Internetworks. *Proceedings of 3rd IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems*, HPCS.
- Nichols, K., Jacobson, V., Zhang, L. (1997) A Two-bit Differentiated Services Architecture for the Internet. *Internet Draft <draft-nichols-diff-svc-arch-00.pdf>*, IETF.

- Nichols, K., Blake, S. (1998) Differentiated Services Operational Model and Definitions. *Internet Draft* <draft-nichols-dsopdef-00.txt>, IETF.
- Roberts, E. (1996) Changing the Lay of the LAN. *Data Communications International*, Issue October.
- Satyavolu, R., Duvedi, K., Kalyanaraman, S. (1998) Explicit rate control of TCP applications. *ATM Forum Document 98-0152R1*.
- Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V. (1997) RTP: A Transport Protocol for Real-Time Applications. *Internet Draft* <draft-ietf-avt-rtp-new-00.ps>, IETF.
- Simcoe, R.J. (1994) Test Configurations for Fairness and Other Tests. *ATM Forum Document 94-0557*.
- Suter, B., Lakshman, T.V., Stiliadis, D., Choudhury, A.K. (1998) Design Considerations for Supporting TCP with Per-flow Queueing. *Proceedings of IEEE INFOCOM '98*, San Francisco.
- XTP-Forum (1995) Xpress Transport Protocol Specification, XTP Revision 4.0. *XTP-Forum Document*.
- Yavatkar, R., Hoffmann, D., Bernet, Y., Baker, F., Speer, M. (1998) SBM (Subnet Bandwidth Manager): A Protocol for RSVP-based Admission Control over IEEE 802-style networks. *Internet Draft* <draft-ietf-issll-is802-sbm-06.txt>, IETF.

## 7 BIOGRAPHY

Matthias Frank graduated in computer science at the University of Paderborn in 1994. His thesis dealt with the design and evaluation of new flow control mechanisms for high speed transport protocols. From 1994 to 1996 he has been with the University of Paderborn working in the European RACE project COMBINE and from 1995 in the European ACTS project AMUSE. In these projects he gained experiences in interactive multimedia system modelling and in the evaluation of service related field trial experiments. In October 1996, he changed to the University of Bonn when his Ph.D. supervisor Prof. Dr. Peter Martini became head of the Institute of Computer Science IV. His current research focuses on end-to-end bandwidth management and control.

Dr. Peter Martini received the M.Sc. and Ph.D. degrees in computer science from the Aachen University of Technology, Germany, in 1986 and 1987, respectively. From 1986 to 1990 he was with the Institute of Computer Science IV at the Aachen University of Technology, Germany, from 1990 to 1996 he was professor of operating systems and computer networks at the University of Paderborn, Germany. Since 1996 Dr. Martini is head of the Institute of Computer Science IV at the University of Bonn, Germany. His research includes performance analysis of heterogeneous networks, protocol specification and implementation, network management, digital libraries, network security and multimedia communication.

# Nondeterministic classifier performance evaluation for flow based IP switching

*Jouni Karvo, Mika Ilvesmäki*

*Helsinki University of Technology*

*P.O.Box 1100, FIN-02015 HUT, Finland*

*Email: {jouni.karvo,mika.ilvesmaki}@hut.fi*

## **Abstract**

In modern IP networks, processing cost in network nodes is considered as a bottleneck. This problem is tackled with traffic based IP switching. The performance of traffic based IP switching depends heavily on flow classification. We demonstrate a method to evaluate the performance gains available with this technique with an optimal nondeterministic classifier giving a practical lower bound for processing cost and compare it with two real life classifiers using recorded traces.

## **Keywords**

IP switching, flow classification, cost optimization, Internet, traffic measurements

## **1 INTRODUCTION**

In modern IP networks, processing resources needed for packet routing are considered as a bottleneck. Several proposals have been made to reduce the load caused by routing for Internet flows in broadband networks. The general method is to label flows and to apply link layer switching functions instead of network layer routing functions. Two main approaches to IP switching exist; these are the flow based IP switching and the topology based IP switching solutions.

### **1.1 IP flow**

IP switching, regardless of the practical realizations, is based on detecting or predicting IP flows. An IP flow is a series of IP packets that share some common properties, such as the IP address prefix, or IP address pair and perhaps also the TCP/UDP port number pair.

Various flow metrics were widely deployed in (Claffy *et al.* 1995, Claffy 1994),

where a flow was defined to be *a series of packets travelling from a constant source to a constant destination*. The definitions of source and destination, and therefore the definition of flow, may be freely chosen to include anything from IP address prefixes defining parts of networks to IP address and TCP port quadruples defining applications. The level of flow definitions is referred to as flow granularity. The higher or finer the granularity the more flows are destined to be created.

## 1.2 IP switching solutions

Traffic based flow switching may assign several flows to one connection. This aims for a more efficient and accurate use of resources. Connections are torn down when a flow timeout occurs because no more packets are being sent on a connection. This timeout value is usually in the order of 30–120 seconds as suggested by Claffy *et al.* (1995). Opinions on whether the traffic based approach scales well when the amount of traffic increases differ (Claffy 1994, Lin and McKeown 1997), but since this issue depends on the connection space available (VC space, in the case of ATM) and on the definition of the flow itself (granularity and timeout values), conclusive statements can not yet be made. Traffic based IP switching is an approach dealing with the problems of short vs. long duration Internet traffic on top of ATM.

The traffic based IP switch is an ordinary routing processor augmented with a flow classifier. The flow classifier detects packet flows and assigns them to their own connections thus reducing the amount of packets that are forwarded through the routing processor. The current available technological solutions for traffic based IP switching use local decision making when determining the flows to be switched. If global decision making is to be used it means using either ATM signalling or RSVP.

Topology switching relies on predefined connections based on routing information and traffic monitoring. A 'topology connection' might include several multiplexed connections to the same part of the network. On the other hand topology based switching might be keeping connections up even if no data is being transmitted. Also the different QoS levels needed require setting up additional paths to a destination with varying QoS characteristics, thus possibly wasting connection space and resources. The issue of aggregated QoS demands needs also further investigation. In a network where switching is based on topology the edge routers recognize aggregated traffic flows which are then switched, rather than routed, through the core network.

The processing capacity available in an IP switch is divided between switched and nonswitched traffic. This division essentially constitutes the process of flow classification. The switched flows can then further be divided to different categories (services) thus providing the possibility for prioritisation and Quality of Service (QoS) for individual flows. The issue of assigning different levels of prioritization or QoS to particular services is not dealt with in this work, but remains as a separate item of research.

### 1.3 Technological issues

A number of different technological solutions have recently emerged. These solutions include Ipsilon's flow based IP Switching (Newman *et al.* 1996b, Newman *et al.* 1997), Cisco's topology based Tag Switching (Rekhter 1997), Toshiba's Cell Switch Router (CSR) (Katsube *et al.* 1996, Katsube *et al.* 1997, Esaki *et al.* 1997) including both the Flow based and the Topology based approach. Other suggestions include Telecom Finland's Switching IP through ATM (SITA) and IBM's Aggregate Route Based IP switch (ARIS). Also the Internet Engineering Task Force's (IETF) Multiprotocol Label Switching (MPLS) workgroup is actively pursuing the subject, with several internet drafts available. In addition, the ATM Forum's MPOA (MPOA 1997) can be viewed as a kind of IP switching solution, although MPOA is argued to be burdened by the heavy load induced by the UNI signalling. The emerged technological solutions, although slightly different from each other, all aim to offer the flexibility of routing combined with the speed, and possibility for QoS, of asynchronous transfer mode (ATM) switching.

The main difference between these solutions, in addition to technical issues and actual implementation, is the way these approaches deal with different levels of traffic aggregation. For instance, Ipsilon's IP switching uses a very fine level traffic granularity defining traffic flows at the finest level of practical use: IP address and TCP port quadruples. On the other hand, for instance Cisco's Tag switching concentrates more on defining and setting up connections based on routing information and aggregated traffic flows from different parts of the network. The MPLS workgroup is currently including both previous views in its plans to introduce IP switching to the Internet. In this article, we concentrate on traffic based flow switching at IP address pair and IP+TCP address/port pair level.

### 1.4 Processing cost of traffic based IP switching

In IP switching, a connection is established for a selected number of flows, that are expected to have enough packets to be carried, so that the cost of the connection is lower than the cost of the routing decisions of individual packets.

Newman *et al.* (1996a) propose an approach where connections are established for certain types of flows, or protocols. We call this kind of criteria *static*. The *dynamic* criteria are based on measuring the packet flows against some criteria to decide when to establish a connection. The decision criteria are called *classifiers*. Lin and McKeeown (1997) compare several classifiers, namely *X/Y Classifier*, *Protocol Classifier* and *Port Classifier*.

The *Port Classifier* is a static classifier based on the assumption that certain types of flows, such as telnet (rlogin) connections are probably longlasting, so a connection is established when the first packet of that kind of connection arrives at the IP switch. The *Protocol Classifier* is also a static classifier that establishes connections on a coarser granularity level (IP address pair). This approach results in lower number

of established connections since several fine granularity level flows are multiplexed to a single connection. The *X/Y Classifier*, presented by Lin and McKeown (1997), is a dynamic classifier that examines the flow and makes a prediction on the future behaviour of the flow according to the past. It requires that  $X$  packets arrive in  $Y$  seconds. An  $X/Y$  classifier with  $Y \rightarrow \infty$  is called a *Packet Count Classifier*, which has also been studied in (Newman *et al.* 1996a, Ilvesmäki *et al.* 1997).

In (Che *et al.* 1997) a dynamic classifier in which the classification criteria are updated based on the resource usage of the IP switch is presented, and in (Ilvesmäki *et al.* 1998) the classification criteria are taught and updated to the system by using a neural network classifier.

Some work has been made in simulating IP switching, see (Lin and McKeown 1997, Ilvesmäki *et al.* 1997, Ilvesmäki and Luoma 1997). The work shows that significant performance gains can be realized using IP switching techniques. In this paper, we propose a *Nondeterministic Classifier* to derive a theoretical value characterizing the maximum attainable gain in processing cost available from IP switching. The *Nondeterministic Classifier* is a classifier that always “guesses” the right answer to the question “To switch or to route” before packets of a flow start to arrive to a node. This way, the classifier is optimal: real life classifiers always lose in processing cost since at least the first packet needs to be inspected and routed regardless of the later decision of switching.

Since our purpose is to illustrate the behaviour of the classifiers and to present a classifier comparison scheme rather than to model traffic, we do not use analytic expressions for the probability distributions associated to the IP flows, but we have used real traces. The traces are drawn from three different types of networks.

In section 2 we develop a model for the nondeterministic classifier. In section 3 we refer shortly to real life classifiers and we use the developed model and real life traces to show the maximal attainable performance gains of IP switching compared to the optimal conditions for port and packet count classifiers. Section 4 gives us a brief summary of the results.

## 2 PROCESSING COST MODEL

In this section, we develop the model for our nondeterministic classifier. A nondeterministic classifier is a theoretical construct that always makes optimal classification decisions, even before seeing the first packet of the flow. We first list our assumptions, then describe the processing cost calculation and finally present the nondeterministic classifier.

### 2.1 Assumptions

In this paper, we limit ourselves to the case where the decision on connection establishment does not affect the traffic demand, for the simplicity of the model. If TCP timeouts would be tight, this might not be the case, since routing — taking more time

than switching — might introduce longer queueing delays that result in misordered or resent packets.

Another assumption taken is that packet labelling, i.e. assigning incoming packets in the already established connections, on the edge of the IP switching network takes approximately the same processing cost as packet routing. We assume that the connection establishment decision is made according to the same criterion on the route of the flow, and that connection establishment takes approximately the same processing cost independently of the initiating node of the establishment in the network. With these assumptions, we may conclude, that by optimizing the processing cost of a single node inside the network, we optimize the whole network behaviour.

The classification of the packets requires processing cost that can be associated in the routing cost of the packets. Another possibility is to assume that flow classification neither with the nondeterministic classifier nor with any real life classifier does require processing capacity. Since our goal is to derive a practical lower bound for processing cost, this assumption is safe.

The background processing load of an IP switch, such as keeping up the links and logging events, is assumed not to depend on the routing or switching decisions. We also assume that flows through our IP switch are independent, which leads us to the conclusion that by minimizing the processing cost of all separate flows results in minimisation of the processing cost of the whole switch.

The number of connections that the network node is able to support is limited. The number of simultaneous connections is in some networks, such as an office's LAN, fairly small, and in some other networks, such as backbone networks of large systems rather high (Lin and McKeown 1997). We assume that the connection table in our systems is sufficiently large. Again, this assumption is safe: if a connection cannot be established when needed due to insufficient resources, total processing cost increases.

We assume that the processing cost of a connection is independent of the state of the system and the length of the flow. I.e. we do not explicitly consider the cost of maintaining the connection, and assume that the results are not affected significantly by this. Note further that omitting the connection maintenance cost lowers the calculated cost which further stresses the performance bound nature of our calculations. Thus our calculations represent the minimum cost attainable for the system in that sense.

## 2.2 Model

Let  $\bar{c}_p$  denote the expected cost of routing one packet, and  $\bar{n}_p$  the expected number of packets in each flow. Let  $\bar{c}_c$  denote the expected cost for connection establishment. Note that packets on an already established connection do not incur processing cost.

Let

$$c = \frac{\bar{c}_c}{\bar{c}_p} \tag{1}$$

denote the relative cost of establishing a connection for a flow to the cost of routing each packet. For example, if we would say “Establishing a connection is 15 times as expensive for the processor as routing a packet”, we would choose  $c = 15$ .

The processing cost is evaluated using traces. Let  $n_r$  denote the number of packets in the trace that are routed, and  $n_c$  the number of connections established. The values of  $n_r$  and  $n_c$  depend on the decisions made by the classifier used. The normalized processing cost is then calculated as follows:

$$C = \frac{1}{n_p} (n_r + c \cdot n_c), \quad (2)$$

where  $n_p$  is the total number of packets in the trace. The processing cost is normalized with  $n_p$  to give more easily understandable results.

### 2.3 Nondeterministic classifier

A nondeterministic classifier is a classifier that knows in advance how many packets would be carried on a flow. I.e., it guesses always correctly (minimizing the processing cost) whether a flow should be switched or routed. We are not able to build such classifiers. All classifiers behave worse than this classifier, thus our nondeterministic classifier represents a goal of performance gain towards which we may compare our classifiers. We also may check whether for our traffic the performance gain attainable by IP switching would be feasible.

When we want to minimize the processing cost of an IP switch, we would establish a connection for all flows, for which  $\bar{c}_c < c_r$ , where  $c_r$  is the processing cost for routing all the packets of the flow, i.e.  $c_r = n_p \cdot \bar{c}_p$ , where  $n_p$  denotes the number of packets in the flow. Thus, the lowest processing cost is achieved when  $c < n_p$ , or the flows that have more packets than  $\lfloor c \rfloor$  are switched and other flows are routed. Here,  $\lfloor c \rfloor$  denotes the greatest integer smaller than or equal to  $c$ . Finally, let  $f(n)$  be the discrete probability distribution function for the flow length:  $f(n) = P[\text{“the flow is } n \text{ packets long”}]$ . The normalized processing cost  $C$  from equation (2) when nondeterministic classifier is used is then

$$C = \frac{1}{n_p} \left( \sum_{n=0}^{\lfloor c \rfloor} n f(n) + c \sum_{n=\lfloor c \rfloor+1}^{\infty} f(n) \right), \quad (3)$$

where  $n_p$  is the total number of packets in the trace.

## 3 REAL LIFE CLASSIFIERS

The static classifiers are based on a predefined set of attributes, such as IP address and port pair. When a packet with matching attributes arrives, the connection is es-



**Table 1** Traces used in this work

<i>Name</i>	<i>Location</i>	<i>Length</i>	<i>Nr of packets</i>	<i>Media</i>	<i>Other information</i>
tct	HUT/LAN 17.6.1997 9.50am	1 hr	142968	Ethernet	-
ebb	HUT/Campus Area Network 29.5.1997 9.00am	1 hr	1107188	10 Ethernet	-
dec	Digital's primary Internet access point 10:00, Thu March 9th, 1995	-	4086848	-	WWW- archive*

\*<http://ita.ee.lbl.gov/html/contrib/DEC-PKT.html>

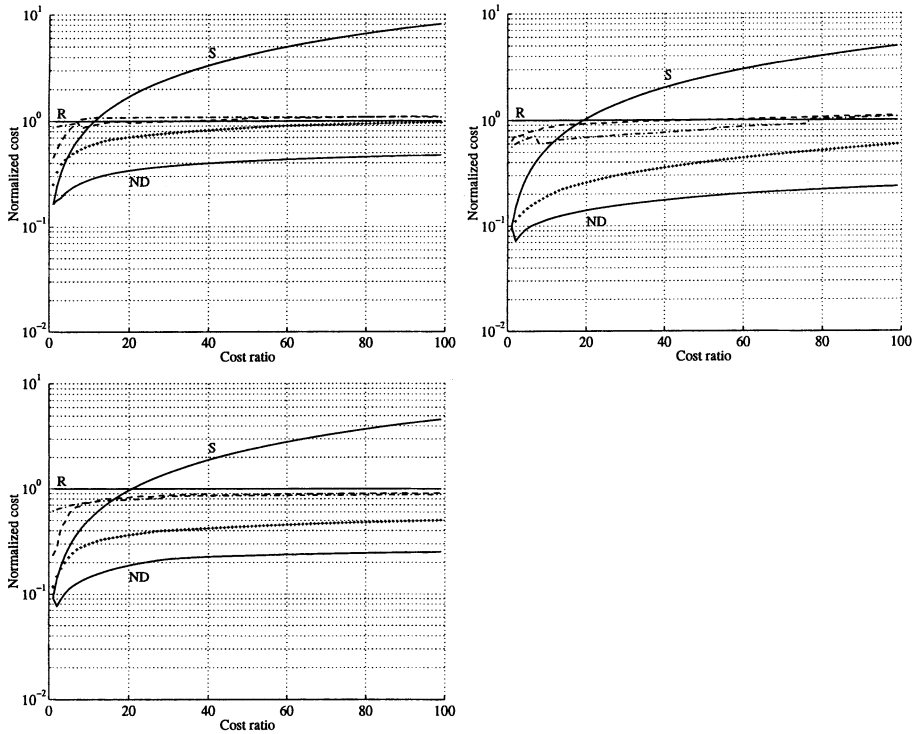
tablished. This means that we shift from the flow per flow classification done by the nondeterministic classifier to flowtype per flowtype classification, which introduces performance loss. Also, the first packet needs processing effort which further downgrades performance.

The dynamic classifiers are based on measuring IP flows and making decisions on connection establishment based on measurement information. Since measurement and routing compete for the same processing capacity, we need to have simple measurement data. This kind of data is e.g. the number of packets sent on a flow, or the number of packets sent in  $Y$  last seconds.

To establish a dynamic classifier, we need to find a condition that describes the state of the traffic process, and is easily measured in a real life IP switch. When the classifier notes that the condition is true, it establishes a connection. For packet count classifier, the condition is defined as  $n_a > n_x$  where  $n_a$  denotes the number of packets received on the flow and  $n_x$  a threshold value. The packet count classifier is based on the assumption that the probability distribution of packet number in the flows is heavy tailed.

We investigated the behaviour of traffic based IP switching using our nondeterministic classifier and two real life classifiers: the port classifier and the packet count classifier. While earlier studies, see e.g. Lin and McKeown (1997), have used predetermined values for the classifiers, we estimated the optimal values for the classifier parameters from traces. Three traces were used (table 1). The traces represent a small IP Local Area Network (tct), a department wide Campus Area Network (ebb) and a major Internet access point (dec). Teardown time of 60s was used in all calculations.

The performance of the classifiers was analyzed using values of cost ratio  $c$  from 1 to 100. The performance of the nondeterministic classifier was analysed adding the



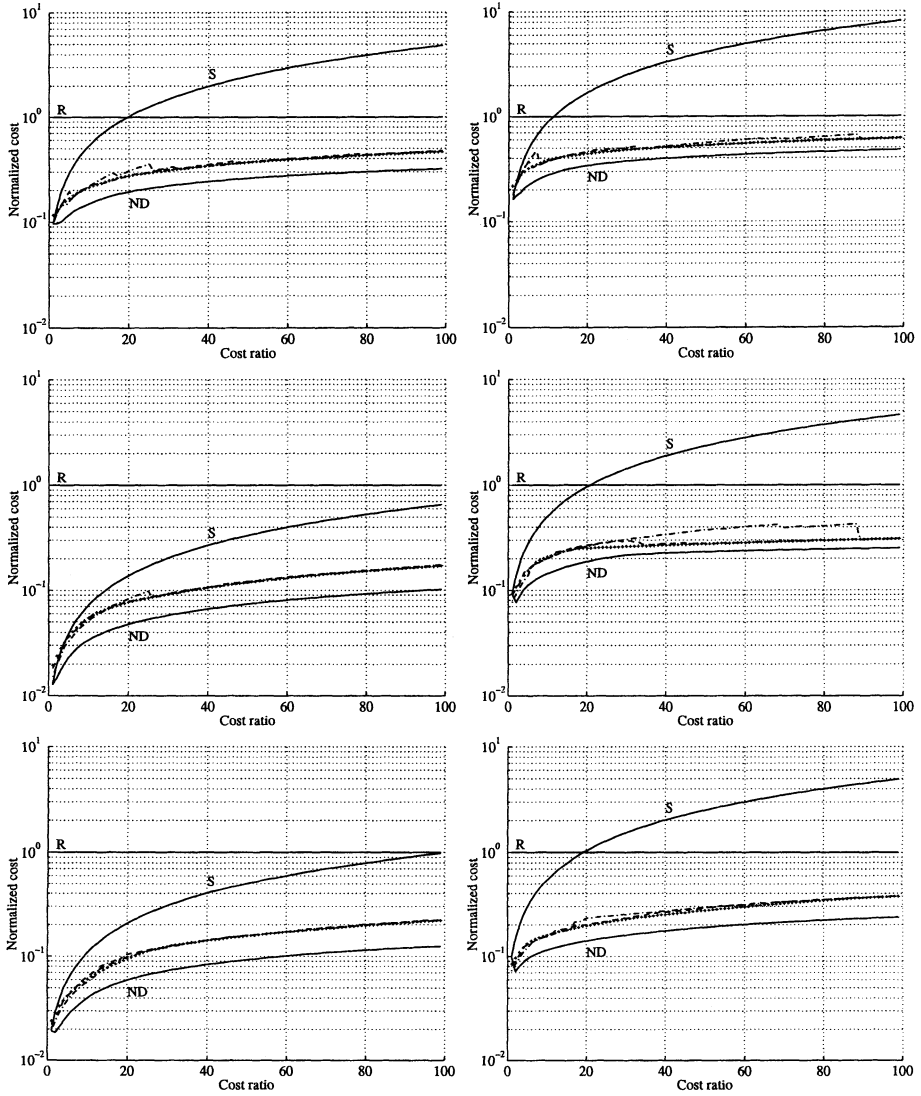
**Figure 1** Normalized processor cost  $C$  with port classifier. Top left: dec trace, top right: tct trace, bottom: ebb trace. Y axes: the total normalized processing cost  $C$ , X axes: the cost of connection establishment vs. cost of packet forwarding,  $c$ . The line R represents the cost of routing all packets, line S the cost of establishing a connection for each flow, and the line ND the cost for the nondeterministic classifier.  $\cdots$  the cost for the port classifier with port list created by the corresponding trace, and  $- \cdot -$ ,  $--$  the costs for the port classifiers with port list created with other traces.

routing cost for all the packets for flows that have less packets than the value of  $c$  and connection establishment cost for all other flows.

Three real life classifiers of each type were created off line using one trace as input for classifier parameter estimation. The resulting classifiers were evaluated using all three traces in turn as input, and the results are presented in figures 1 and 2.

Figures 1 and 2 show the lowest achievable normalized processing cost as the cost for the nondeterministic classifier. IP switching seems to offer even 90 % reduction to the routing cost as an upper limit. Even if the cost ratio  $c$  grows to high, maybe even slightly unrealistic, values a cost reduction of 50 to 60 % is possible in theory.

The port classifier parameters are estimated by grouping together all flows with the same IP address pairs and source port numbers. The mean remaining flow length after receiving the first packet is calculated for each group. For the flow groups that



**Figure 2** Normalized processor cost  $C$  with packet count classifier. Top: dec trace, middle: tct trace, bottom: ebb trace. Left: flow granularity at IP address level, right: flow granularity at IP address and port level. Y axes: the total normalized processing cost  $C$ , X axes: the cost of connection establishment vs. cost of packet routing,  $c$ . The line R represents the cost of routing all packets, line S the cost of establishing a connection for each flow, and the line ND the cost for the nondeterministic classifier.  $\cdots$  cost for the packet count classifier with parameter  $n_a$  estimated from the corresponding trace, and  $-\cdot-$ ,  $--$  the costs for the packet count classifiers with  $n_a$  estimated from other traces.

have the mean flow length greater than the value of the cost ratio  $c$ , the classifier should establish a connection for all flows belonging to the group. For the flow groups that have the mean flow length smaller than the value of the cost ratio  $c$ , all packets of the flows in the group should be routed. Thus the required parameters for the port classifier is the list of the port pairs and source and destination address pairs for which connections are established. After it has been found the classifier is ready for classification. When distinguishing flows, we used IP address pairs, and source and destination port pairs. Figure 1 shows the results for port classifier.

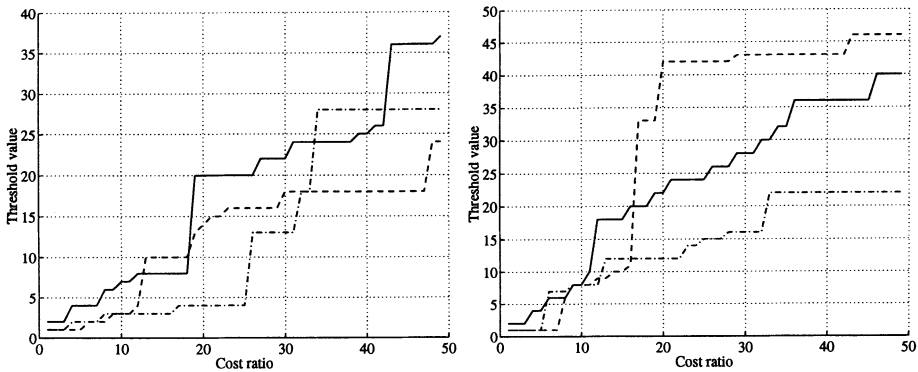
The real life port classifier seems to be capable at the best to only 50 to 60 % improvements. Figure 1 shows that the port classifier performs adequately only in the same or very similar network than with which it has been created. When the classifier is used in another network its performance degrades significantly (for our traces, to only 10 to 30 % improvements). The processing cost may in some cases even be higher than for pure routing. In different parts of the network there exist different kinds of traffic profiles. Consequently, when using a port classifier one needs either to determine the switched ports separately for each network or to use a system that can adapt to the changes in the traffic profile. More generally, it can be said that implementing any kind of consistent Quality of Service on a connection to the Internet is difficult, since different services require different handling on different parts of the network.

The parameter for the packet count classifier is estimated by calculating the processing cost if  $n_a$  first packets of the flows would be received and after that the connections would be established. Then the value of  $n_a$  for which the cost attains its minimum is chosen for the threshold value of the classifier, i.e. minimize

$$C = \frac{1}{n_p} \left( \sum_{n=0}^{n_a} n f(n) + \sum_{n=n_a+1}^{\infty} (n_a + c) f(n) \right), \quad (4)$$

subject to  $n_a \geq 1$ , where  $f(n)$  is the flow length distribution. The parameter attained is the threshold value  $n_a$ . When working, the classifier decides to establish a flow when the number of received packets exceeds the threshold value. We calculated the processing cost with two levels of granularity: first with IP address pairs, then with IP address pairs and source and destination port pairs. The optimal thresholds found are shown in figure 3 and the costs in figure 2

Figure 3 shows that as the cost ratio grows the packet count threshold value also grows. This is expected, since the higher the packet count threshold value is the less connections are set up. This behavior minimizes quite effectively the cost of packet count classifier as seen in figure 2. The cost of packet count classifier follows the nondeterministic classifier cost with a slightly worse performance. The maximum cost reductions with the packet count classifier are in the 90 % level, and even for high practical cost ratios the classifier has relatively reasonable levels of processing cost reduction (about 40 to 70 %). The results also show that the cost reduction is rather insensitive to the selection of the packet count threshold value, which suggests that the same packet count classifier can be used in a variety of networks.



**Figure 3** Optimal threshold values used for packet count classifier. Left: IP address level granularity, right: IP address and port level granularity. — dec trace, — — ebb trace, — · — tct trace.

## 4 CONCLUSIONS

The flow classifier has a significant impact on IP switching system performance. In earlier works different classifiers were compared but no bounds were present for the maximal performance gains available. In this paper, we proposed a nondeterministic flow classifier for performance evaluation of flow based IP switching solutions. The nondeterministic classifier is an optimal classifier with which other classifiers can be compared.

As an example, we evaluated the performance of two real life classifiers by a comparison to the nondeterministic classifier. The results show that even 90 % of processing cost reductions may be achieved by flow based IP switching in theory. The practical classifiers perform worse.

The port classifier offers moderate cost reductions but is heavily dependent of the underlying traffic. The packet count classifier offers a good performance and is less sensitive to the underlying traffic. Neither of these classifiers supports implementing QoS classes for flows well.

We claim that the nondeterministic classifier based performance comparison is widely applicable and relatively easy to implement. Furthermore, it gives comprehensible results.

## ACKNOWLEDGEMENTS

The authors wish to thank Prof. Jorma Virtamo and Prof. Gunnar Karlsson for their comments on the early stage of the work. The research was funded by TOVE and IPANA projects at the Helsinki University of Technology.

## REFERENCES

- MPOA (1997) *Multiprotocol over ATM Version 1.0*. ATM Forum, 07.04.1997.
- Che, H., Li, S.-Q., and Lin, A. (1997) Adaptive resource management for IP/ATM hybrid switching systems. In *Broadband Networking Technologies* November 1997, Civanlar S., and Widjaja I., Eds., **3233**, SPIE, 328–339.
- Claffy, K.C. (1994) *Internet Traffic Characterisation*. Ph.D. thesis, University of California, San Diego.
- Claffy, K.C., Braun, H.-W., and Polyzos, G.C. (1995) A parametrizable method for internet traffic flow profiling. *IEEE Journal on Selected Areas in Communications* **13**(8) 1481–1494.
- Esaki, H., Matsuzawa, S., Mogi, A., Ichi Ngami, K., Jinmei, T., Kon'no, T., Katsube, Y. (1997) Cell switch router (csr) – label switching router supporting standard atm interfaces. In *Broadband Networking Technologies* November 1997, Civanlar S., and Widjaja I., Eds., **3233**, SPIE, 2–10.
- Ilvesmäki, M., Kilkki, K., and Luoma, M. (1997) Packets or ports – the decisions of IP switching. In *Broadband Networking Technologies* November 1997, Civanlar S., and Widjaja I., Eds., **3233**, SPIE, 53–64.
- Ilvesmäki, M., and Luoma, M., (1997) IP switching in a simplified ATM environment. In *Broadband Networking Technologies* November 1997, Civanlar S., and Widjaja I., Eds., **3233**, SPIE, 65–76.
- Ilvesmäki, M., Luoma, M., and Kantola, R. (1998) Flow classification schemes in traffic based multi-layer IP switching – comparison between conventional and neural approach. Accepted to *Computer Communications* **22**.
- Katsube, Y., Nagami, K.-I., and Esaki, H. (1996) *Cell switch router – Basic concept and migration scenario*. Toshiba R&D Center.
- Katsube, Y., Nagami, K., and Esaki, H. (1997) *Toshiba's router architecture extensions for ATM: Overview*. Toshiba R&D Center.
- Lin, S., and McKeown, N. (1997) A simulation study of IP switching. In *Proceedings of ACM SIGCOMM* September 1997.
- Newman, P., Lyon, T., and Minshall, G. (1996a) Flow labelled IP: A connectionless approach to ATM. In *IEEE INFOCOM Joint Conference on Computer Communications* San Francisco, California, March 1996, **3**, 1251–1260.
- Newman, P., Lyon, T., and Minshall, G. (1996b) Flow labelled IP: Connectionless ATM under IP. In *Network & Interop*, Las Vegas, April 1996.
- Newman, P., Minshall, G., and Lyon, T. (1997) IP switching: ATM under IP. [www.ipsilon.com](http://www.ipsilon.com), January 1997.
- Rekhter, Y. (1997) Tag switching architecture – overview. In *Broadband Networking Technologies* November 1997, Civanlar S., and Widjaja I., Eds., **3233**, SPIE, 11–19.

## **Part Eleven**

---

# **QoS Routing and Scheduling**

# Internet QoS Routing using the Bellman-Ford Algorithm

*D. Cavendish and M. Gerla*

*Computer Science Department,*

*University of California at Los Angeles*

*405 Hilgard Avenue, Los Angeles, CA 90024-1596, USA*

*{dirceu,gerla}@cs.ucla.edu*

## Abstract

Multimedia applications are Quality of Service (QoS) sensitive, which makes QoS support indispensable in high speed Integrated Services Packet Networks (ISPN). An important aspect is QoS routing, namely, the provision of QoS routes at session set up time based on user request and information about available network resources. This paper develops optimal QoS routing algorithms within an Autonomous System (AS). Previous approaches have been based either on minimizing a **single** metric (delay, for instance) or a combination of multiple metrics, optimizing one at a time, in a hierarchical fashion. Our approach finds minimum hop paths which satisfy **multiple** QoS constraints. We argue that a QoS version of the Bellman-Ford routing algorithm provides the best strategy for QoS routing problems of a given type. We show that Bellman-Ford is very powerful in solving most multiple constrained routing problems arising in a flat network (within an autonomous system), if the minimum hop is the main objective function. We further illustrate the importance of Bellman-Ford QoS routing algorithms with regard to network utilization and session blocking through simulation experiments.

## Keywords

Internet Routing, Quality of Service Constraints

## 1 INTRODUCTION

Quality of Service (QoS) sensitive applications are becoming popular, as Internet users move to more demanding applications with regard to the types of service requested from a packet network. Thus, QoS support for these applications is necessary to meet oftentimes stringent end-to-end requirements such as bandwidth, delay, and packet loss. The support of QoS applications in a packet network includes a broad range of functions such as priority mechanisms for flows, scheduling disciplines, traffic shaping schemes, and routing algorithms. This paper is concerned with the last issue, i.e., the provision of



routing algorithms capable of finding routes which comply with QoS applications requirements.

Routing QoS sensitive applications consists in finding a path which complies with end-to-end constraints derived from the QoS users needs. This task translates into finding routes in a network scenario of multiple metrics. In general, routing optimally a flow with such end-to-end constraints (multiple metrics) is known to be an NP-complete problem (Garey *et al.* 1979). Recent papers have addressed the QoS constrained routing problem using a variety of strategies. Some researchers have proposed approximate solutions based on minimizing a single objective function at a time (i.e., defining a hierarchy of metrics) (Guerin *et al.* 1997a) for a generic multiple metric routing problem. Others have succeeded in reducing the problem complexity to a polynomial degree by assuming a particular scheduling discipline and traffic shaping policy at each router ((Pornaivalai *et al.* 1997) (Zhao *et al.* 1997)). Our work somehow lies in between these two approaches, as it makes certain assumptions about the multiple metrics which define the problem, but not about specific router behavior in handling the traffic flow.

In considering routing with multiple constraints, it has been noticed that the Bellman-Ford (BF) algorithm can potentially solve a two metric routing problem, when one of the metrics is the hop count (Guerin *et al.* 1997a). This fact favors the Bellman-Ford algorithm when compared with the Dijkstra algorithm, which lacks this capability. This is because Dijkstra algorithm does not search for a minimum path cost in ascending order of number of hops, as Bellman-Ford does. We have decided to investigate further the Bellman-Ford algorithm's ability to solve multiple constrained routing problems. The use of a minimum hop routing strategy is justified since it is likely to provide the lowest session rejection probability for a given network load, if no knowledge about the distribution of connections is assumed. Note that QoS applications usually require end-to-end **bounds** with regard to various metrics, rather than the minimization of a specific end-to-end metric (e.g., delay). We have chosen to find the path with minimum number of hops among all which satisfy the multiple end-to-end constraints.

The paper is organized as follows. Section 2 describes the model for a flat (AS domain) network model with multiple metrics. It also describes related work in the QoS routing area. In section 3, we define a Bellman-Ford algorithm capable of handling multiple cost metrics. We show some routing problems that can be solved exactly and others which cannot be solved by this enhanced Bellman-Ford algorithm. Section 4 explains how bandwidth, buffer, delay, delay jitter, and packet loss metrics can be handled by a Bellman-Ford algorithm, in a typical QoS supportive packet network. In section 5, we exemplify the use of a multiple metric Bellman-Ford algorithm in supporting QoS flows, comparing it with other popular routing strategies. Section 5 concludes the paper and points to future research directions.

## 2 QOS ROUTING WITH MULTIPLE CONSTRAINTS

We model a network as a directed graph  $G(V, E)$ , where  $V$  is the set of nodes ( $|V| = N$ ) and  $E$  the set of directed *edges*. Every edge  $(i, j) \in E$  is associated with a set  $\mathcal{M}$  of non-negative values, referred to as edge metrics. Thus an edge  $(i, j)$  has cost  $m(i, j)$  with respect to metric  $m \in \mathcal{M}$ . Examples of edge metrics are: edge cost (c), bandwidth (b), delay (d), delay jitter (j), loss probability (l).

A path  $Pa(s, t)$  is a sequence of vertices  $v_s, \dots, v_i, v_{i+1}, \dots, v_t$  such that  $\forall s \leq i \leq t$ , edge  $(v_i, v_{i+1}) \in E$ .

Upon each metric  $m \in \mathcal{M}$ , we further define the cost  $c_m(s, t)$  of a path  $Pa(s, t)$  as a generic function  $f$  of the path's edges metrics, or:

$$c_m Pa(s, t) = f(m(v_s, v_{s+1}), \dots, m(v_i, v_{i+1}), \dots, m(v_{t-1}, v_t)) \quad (1)$$

Examples of path cost functions are:

$$c_{bw} Pa(s, d) = \min_{(i,j) \in Pa(s,d)} b(i, j) \quad < \text{bandwidth} > \quad (2)$$

$$c_{bf} Pa(s, d) = \min_{(i,j) \in Pa(s,d)} bf(i, j) \quad < \text{buffer} > \quad (3)$$

$$c_h Pa(s, d) = \# \text{edges of } Pa(s, d) \quad < \text{hop count} > \quad (4)$$

$$c_d Pa(s, d) = \sum_{(i,j) \in Pa(s,d)} d(i, j) \quad < \text{delay} > \quad (5)$$

$$c_j Pa(s, d) = \sum_{(i,j) \in Pa(s,d)} j(i, j) \quad < \text{jitter} > \quad (6)$$

$$c_l Pa(s, d) = 1 - \prod_{(i,j) \in Pa(s,d)} [1 - l(i, j)] \quad < \text{packet loss} > \quad (7)$$

where  $b(i, j)$ ,  $bf(i, j)$ ,  $d(i, j)$ ,  $j(i, j)$ ,  $l(i, j)$  are the bandwidth, buffer, delay, delay jitter, and packet loss probability of link  $(i, j)$ , respectively \*.

A QoS routing task consists of finding a path  $Pa(s, t)$  suitable to a given application which has end-to-end constraints. These constraints translate into specific path value bounds for one or more of the metrics defined above. According to the definitions above, one can see that bandwidth and buffer are minimum path constraints, while delay and delay jitter are additive constraints, and finally loss is a multiplicative constraint. In particular, we are interested in the "best" path which satisfies a given application's end-to-end constraints, in the sense that such path should use the minimum amount of

---

\* We assume that a QoS application requires a minimum amount of buffering at each router along its path

network resources. Therefore, an optimal path is generally defined as a path with minimum number of hops which still satisfies a given set of metric (upper/lower) bounds.

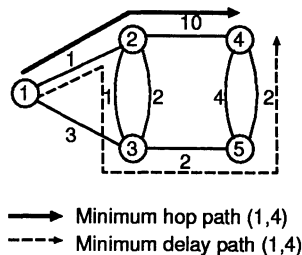
## 2.1 Related work

The most general shortest path problem with multiple constraints is known to be NP-Complete (Garey *et al.* 1979). Recent research in the solution of Bellman's Equations with multiattributes (Henig 1994) establishes some conditions on the set of vectors of path values so that an order of preference among the possible paths can be defined, and thus a solution obtained. In general, however, the checking of those conditions on a particular instance of the problem takes exponential time. (Henig 1994) also shows that, for a given class of multiattribute shortest path problems, a polynomial solution is available. Our work addresses a small sub-class of the above mentioned class.

(Wang *et al.* 1995) first modelled the QoS routing problem as a multiple metric shortest path problem. By using standard reduction techniques, they showed that a routing problem subject to any two metrics among cost, delay, loss probability and jitter, is NP-Complete. However, they make no assumptions about the multiple metrics when deriving the impossibility results. Since then, solutions for multiple constrained routing problems have been reduced to either the minimization of a single cost metric and the verification that the path computed satisfies the required bounds (Lee *et al.* 1997); or the definition of a hierarchy of cost functions, to be optimized one at a time (Guerin *et al.* 1997a). Our approach differs from previous research work in that: i) It insists in computing **minimum** hop paths which satisfy multiple metric cost **bounds**; ii) it makes realistic assumptions about delay, loss probability, jitter, and bandwidth metrics within an AS.

## 3 USING BELLMAN-FORD ALGORITHM WITH MULTIPLE METRICS

Consider the network shown in Figure 1. Arc values represent link delays.



**Figure 1** Delay-hop routing example

Let us assume we wish to compute a path between nodes 1 and 4, subject to delay bound,  $D$ . We look for the minimum hop path with delay smaller or equal to  $D$ . Our choice of paths will obviously depend on the value of  $D$ : if  $D > 11$ , the minimum hop path shown in the figure should be used; if  $7 \leq D < 11$ , path (1,3,5,4) should be used; if  $D = 6$ , the minimum delay path shown in the figure is the only alternative. For delay requirements smaller than  $D = 6$ , no path is available.

The original Bellman-Ford (BF) algorithm was designed to compute shortest paths between a given vertex  $s$  and multiple destinations. It did not include multiple metrics. Moreover, path cost function was always considered to be of additive nature, as in the path delay cost above. We wish to extend BF algorithms to handle as many metrics of the types previously defined as possible.

Without loss of generality, we assume vertex 1 as the vertex from which we wish to compute distances to all other network vertices. As usual in BF algorithm, we define  $D_i^h$  as the minimum distance with respect to some metric  $d$  between vertex 1 and vertex  $i$  with at most  $h$  number of hops. The Bellman-Ford equation (Cormen *et al.* 1990) is:

$$D_i^{h+1} = \min_{j \in N(i)} [d(i, j) + D_j^h], \quad \forall i \neq 1 \quad (8)$$

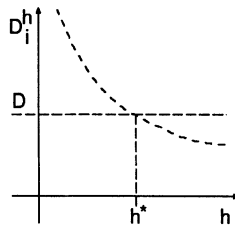
where  $N(i)$  is the set of neighbors of vertex  $i$ .

Starting with initial conditions -  $D_i^0 = \infty$  and  $D_1^h = 0$  - BF algorithm iterates eq. (8) until a predefined number of hops  $H_{max}$  has been reached,  $H_{max} \leq N$  (If the shortest possible path is sought, regardless of number of hops,  $H_{max} = N$ ).

In a regular BF algorithm, it is well known that:

**Theorem 1** (Bertsekas *et al.* 1992)  $D_i^h$  is non-increasing with  $h$ , regardless of the cost metric  $d$  used.

Essentially, the non-increasing property of the distance  $D^h$  is provided by the **min** operator of eq. (8). Notice also that the search direction defined by the **min** operator is the steepest descend of  $D^h$ . Figure 2 illustrates  $D_i^h$  dependence on  $h$ .



**Figure 2** Non-increasing cost function

The first routing problem of interest is:

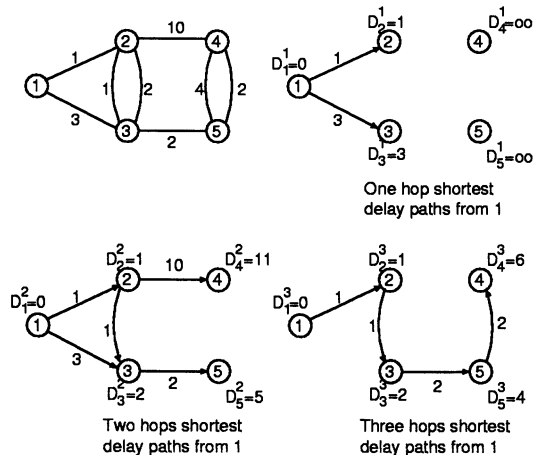
**Problem 1** Find a minimum-hop route  $Pm(s, t)$  with  $d$  (delay) cost upper bound  $c_d Pm(s, t) \leq D$

It is not difficult to see that we can use BF algorithm to solve problem 1. More specifically, if  $h^*$  is defined to be the shortest number of hops in which  $D_i^h \leq D^*$ , then:

**Theorem 2** A regular BF algorithm outputs the minimum-hop path with  $c_d Pa(s, t) \leq D$  at the first time  $D_i^h \leq D$ .

**Proof of Theorem 2** We run BF algorithm until the first time  $D_i^h$  falls below  $D$ , say at  $h^*$ . Since the number of hops always increases as BF progresses, and by assumption  $h^*$  is the point in which  $D_i^h$  falls below  $D$  for the first time, the Theorem is proven (Fig. 2).

Figure 3 illustrates successive iterations of the Bellman-Ford algorithm for the example shown above. It is worth noting that the Bellman-Ford algorithm has a worst case complexity of  $O(N^3)$  (Bertsekas *et al.* 1992), which is higher than the Dijkstra algorithm complexity of  $O(N \log N)$ . However, the regular Dijkstra does not classify the paths it generates on the basis of hop count, as BF algorithm does. This feature comes handy when dealing with multiple constrained routing problems.



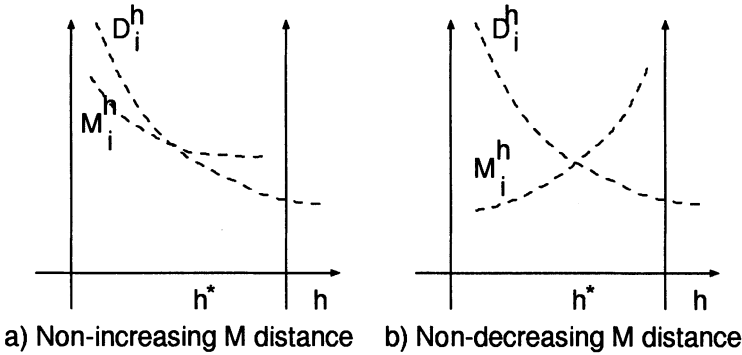
**Figure 3** Successive iterations of Bellman-Ford Algorithm

\*We will use \* to denote optimal paths

We now introduce a second distance  $M$  in the BF algorithm, in the following way:

$$M_i^{h+1} = f[m(i, k), M_k^h], \quad k \text{ s.t. } \min_{j \in N(i)} [d(i, j) + D_j^h] \quad (9)$$

i.e.,  $M_i^h$  is the path distance with regard to metric  $m$  of the path chosen in order to minimize the distance  $D_i^h$ . For example, metric  $m$  could be any of the metrics defined in Eqs. (2) through (7). Notice that a priori we can not claim any property of  $M_i^h$ , as stated in Theorem 1 for  $D_i^h$ , unless we assume some correlation between metrics  $m$  and  $h$ . If such a correlation exists, two cases are of interest:  $M_i^h$  is non-increasing with  $h$ ;  $M_i^h$  is non-decreasing with  $h$ . Figure 4 illustrates such cases.



**Figure 4** Multiple cost functions

For both cases, we further assume that  $M_i^h$  has no discontinuities, namely, any two paths with the same number of hops has the same distance  $M$ . So vertical jumps on Figure 4 are forbidden. Given that the path cost functions dealt with are of the form depicted in Fig. 4, one can use the same strategy as before to compute minimum hop paths with bounds on multiple constraints. More specifically, we define the following routing problems:

**Problem 2** Find a minimum-hop route  $P_m(s, t)$  with  $d$  cost upper bound  $c_d P_m(s, t) \leq D$  and  $m$  cost upper bound  $c_m P_m(s, t) \leq M$ .

**Problem 3** Find a minimum-hop route  $P_m(s, t)$  with  $d$  cost upper bound  $c_d P_m(s, t) \leq D$  and  $m$  cost lower bound  $c_m P_m(s, t) \geq M$ .

It is not difficult to see that:

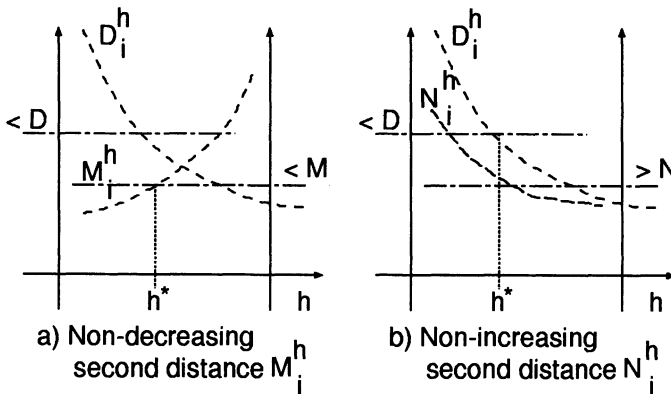
**Theorem 3** A regular BF algorithm outputs the minimum-hop path with  $c_d Pa(s, t) \leq D$  and  $c_m Pm(s, t) \geq M$  if at the first time the condition  $(D_i^h \leq D)$  we have  $(M_i^h \geq M)$  is true for a non-increasing  $M^h$  distance. If it so happens that  $(M_i^h \leq M)$ , the problem has no solution.

**Proof of Theorem 3** Follows from the non-increasing property of distance  $M^h$  and Theorem 2.

**Theorem 4** A regular BF algorithm outputs the minimum-hop path with  $c_d Pa(s, t) \leq D$  and  $c_m Pm(s, t) \leq M$  if at the first time the condition  $(D_i^h \leq D)$  we have  $(M_i^h \leq M)$  is true for a non-decreasing  $M^h$  distance. If it so happens that  $(M_i^h \geq M)$ , the problem has no solution.

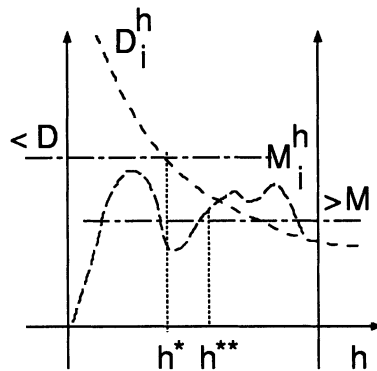
**Proof of Theorem 4** Follows from the non-decreasing property of distance  $M^h$  and Theorem 2.

Figure 5 illustrates two metric constrained routing problems and their solutions.



**Figure 5** Two constrained BF solvable routing problems

Notice that if we are seeking an upper bound for a non-increasing distance  $N_i^h$ , or a lower bound for a non-decreasing distance  $M_i^h$ , the doubly constrained minimum hop problem can not be solved by the Bellman-Ford algorithm. Also, the monotonic non decreasing/increasing properties of the distances  $M_i^h$  and  $N_i^h$  are key to claim that the problem has no solution if the two bounds are not met. For instance, let's assume two distances  $D_i^h$  and  $M_i^h$ , based on generic metrics  $d$  and  $m$ , whose dependency with the number of hops is displayed in Figure 6.



**Figure 6** When BF-algorithm fails

The path represented by  $h^*$  satisfies the upper bound  $D$ , although it violates the lower bound  $M$ . However, according to the figure, it is possible to continue lowering the distance  $D_i^h$  and find a path with  $h^{**} > h^*$  satisfying the lower bound  $M$ . Notice that we can not claim that the path determined by  $h^{**}$  is the shortest one (in number of hops) sought satisfying both bounds. This is so because, once the upper bound  $D$  is satisfied, a search on the steepest descend direction on distance  $D_i^h$  might no longer be the best search strategy towards finding a minimum hop path respecting the bound  $M$ . Conversely, although a search on the steepest descend direction on distance  $M_i^h$  will lead to a shortest path on  $h$  satisfying the lower bound  $M$ , this search strategy is not guaranteed to yield a monotonic decrease on the distance  $D_i^h$  (see the definition of distances  $D_i^h$  and  $M_i^h$ ).

Now for the main theorem. For a multiple constrained routing problem, define distance  $D_i^h$  as in eq. (8) wrt a given metric  $d(i, j)$ . Then:

**Theorem 5** *Partition the multiple constrained minimum hop routing problem into subproblems by pairwising metric  $d$  with each other metric. The original multiple constrained problem can be solved by a multiple metric BF algorithm if and only if each subproblem is of the form defined by either Theorems 3 or 4.*

**Proof of Theorem 5** *If follows from Theorems 3 and 4 and the fact that multiple metrics interact only through the distance  $D_i^h$  which defines the search direction (see eq. 9).*

One more observation is due. For a generic metric, the verification whether a given distance has non-increasing/non-decreasing property might take exponential time (see (Henig 1994) for further discussions on this issue). However, this property is called upon only when the multiple metric BF algorithm fails to find a path, to claim that the problem has no solution. If a path satisfying



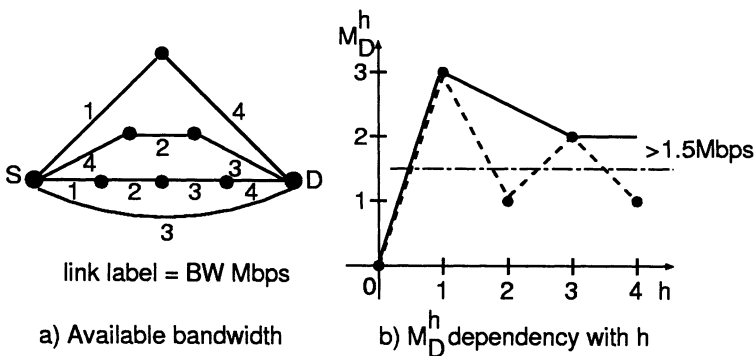
the multiple constraints is found, Theorem 1 implies that this path is indeed the shortest hop path sought.

#### 4 SOLVING ROUTING PROBLEMS WITH MULTIPLE CONSTRAINTS

We now take a closer look at practical metrics defined in section 2, and their dependency on the number of hops in a typical network scenario. Although in establishing the theory we did not assume any particular scheduling discipline exercised at the routers, for realism we draw our examples from recent research work in the provision of delay and delay jitter bounds.

##### Bandwidth

Typically, bandwidth availability is a metric in which we should not expect any correlation with hop count. This is because bandwidth availability in a link depends not only on the distribution of flows but also on the routing strategy utilised to route flows across the network. Figures 7 a) and b) illustrate a network with available bandwidth and path bandwidth dependency on the number of hops, respectively.



**Figure 7** Typical path bandwidth dependency with number of hops

The dashed line shows a typical behavior of the distance  $M_D^h$ . In a search for a QoS path requiring minimum bandwidth, say  $1.5Mbps$ , there is no need in considering paths that fall below a  $1.5Mbps$  threshold line, as shown in the figure. By disregarding all links in the graph with bandwidth less than the required, we reduce our search space to paths with bandwidth at least as much as the requested. This is equivalent to redefining distance  $D_i^h$  as:

$$D_i^{h+1} = \min_{j \in N(i) \& m(i,j) > M} [d(i, j) + D_j^h], \quad \forall i \neq 1 \quad (10)$$

The solid line in figure 7 b) shows the new distance  $M_D^h$  dependency when links with  $1Mbps$  are not considered in the routing problem. Notice that if we were searching for a path whose QoS requirements translates into minimum bandwidth only, the path in which  $M_D^h$  crosses the threshold line for the first time should be used. Notice also that the curves assume  $M_D^0 = 0$  as the initial condition for the routing algorithm.

## Buffer

In general, the amount of buffers allocated to a given session is expected to be related to the bounds on loss probability and delay guaranteed by the router. Buffer requirements per node, thus, depend on the scheduling and service disciplines exercised by the router (see (Zhang 1995) for a survey on relevant service disciplines for deterministic delay bounds). If a given session needs a specific amount of buffers for reasons other than loss bounds (e.g., delay bounds provided by some scheduling discipline (Zhang 1995)), we can use definition 3 for buffer path cost. According to definition 3, it is easy to see that a distance  $B_i^h$  has the same dependance on the number of hops as for the bandwidth case described above. Thus, for a given QoS connection with buffer requirement  $> B$ , a good strategy is to eliminate all links without the minimum buffer storage.

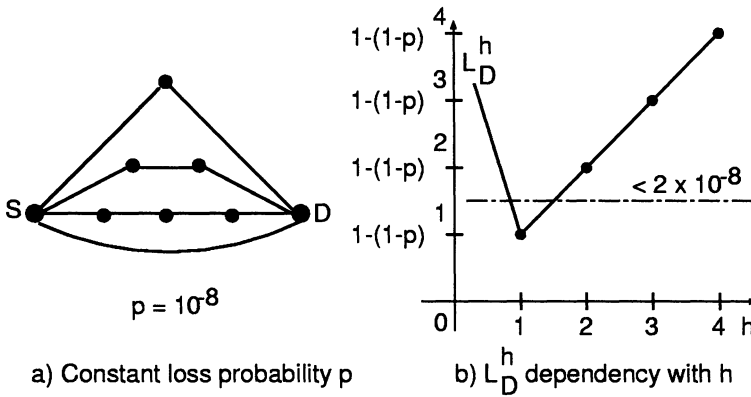
The path cost definition for buffers assumes a uniform buffer requirement along all routers of a given path. Recent scheduling disciplines (Georgiadis *et al.* 1996) imply buffer allocation which grows linearly with the hop count. In this case, we can also redefine distance  $D_i^h$  so as to reduce the number of metrics in the routing problem, as:

$$D_i^{h+1} = \min_{j \in N(i) \& b_f(i,j) > h \times B} [d(i, j) + D_j^h], \quad \forall i \neq 1 \quad (11)$$

Notice that equation (11) is well defined since  $h$  is known at all times of the BF algorithm computation.

## Loss probability

We believe that, at least within an autonomous system (AS), the various routers will typically advertise a constant loss probability for a given class of traffic. This being the case, we should expect that a distance  $L_D^h$  dependency with hop count as exemplified in Figure 8.



**Figure 8** Typical path loss dependency with number of hops

We require  $L_D^0 = \infty$  if there is no zero hop path for the source destination pair  $S, D$ . Notice that as soon as  $L_D^h \neq \infty$ , the behavior of  $L_D^h$  is monotonically increasing (almost linear) with the number of hops. Figure 8 b) shows a QoS loss requirement of  $L < 2 \times 10^{-8}$ . If this is the only QoS requirement, the one hop path in the figure is the answer to our QoS routing problem.

### Delay Jitter

We anticipate that, as in the case of loss probability, all routers of a given AS will advertise the same delay jitter bounds. This is essentially because propagation delays are assumed constant within an AS, which makes delay jitter dependent on the scheduling and service disciplines exercised by the router. For deterministic jitter bounds, the same values are expected to be advertised for a given traffic class (Georgiadis *et al.* 1996) (Zhang 1995). In this case, the dependency of the distance  $J_D^h$  with the number of hops is linear, thus similar to the distance  $L_D^h$  shown above.

### Delay

With regard to hop delays, a delay metric  $d(i, j)$  will have various values for different links. Since  $d(i, j)$  includes link propagation delays plus queueing delay and service time (packet size), even if queue waiting times are similar across routers within an AS, propagation delays and service times vary wildly. Therefore, it is not legitimate to assume any particular dependency behavior of  $D_i^h$  with the number of hops. However, if we define the distance  $D_i^h$  as in eq. (8), Theorem 1 guarantees the non-increasing property of  $D_i^h$  necessary to compute in polynomial time QoS paths with multiple constraints, regardless of the  $d(i, j)$  metric behavior.

From the characteristics of the various metrics of interest and from Theorem 5, it is easy to check that a multiple constrained routing problem constructed with any subset of the metrics discussed above is solvable by a multiple metric

BF algorithm. Notice that the restrictive uniform behavior on loss probability and delay jitter metrics assumed is not strictly necessary for the QoS aware Bellman-Ford algorithm to work, but only that their corresponding distances  $L_i^h$  and  $J_i^h$  be non-decreasing functions with  $h$ . For instance, in Figure 8, if the link of the one hop path has loss probability  $p_1$ , the links of the two hop path have loss probability  $p_2$ , and so on, a QoS aware Bellman-Ford will work provided that:  $1 - (1 - p_1) \leq 1 - (1 - p_2)^2 \leq 1 - (1 - p_3)^3 \leq 1 - (1 - p_4)^4$ . This means that the set of routing problems solvable by the multiple metric Bellman-Ford algorithm defined in the previous section is larger than the set considered here. We have considered uniform link values for these metrics because we can easily check the non-decreasing distances property with the number of hops. For a generic metric, however, this checking takes exponential time (see (Henig 1994) for further discussions on this issue). For generic loss probability and delay jitter metrics, one can still use the multiple metric Bellman-Ford algorithm defined in the previous section. Strictly speaking, the only limitation is that, in case a feasible solution is not found, one can not be sure that indeed such a feasible path does not exist.

## 5 SIMULATION STUDY

In this section, we illustrate the advantages of our QoS routing approach, namely, minimizing hop distance while satisfying various metric bounds. Our figure of merit is session acceptance probability, which is evaluated for various network loads. We assume each session has specified QoS bounds on the metrics considered, possibly conveyed by RSVP signalling upon entering the Autonomous System (AS) cloud (Guerin *et al.* 1997b). Although a session may traverse several ASs, where each AS requires a QoS routing solution of its own, we consider a single AS only. Therefore, our bounds do not represent end-to-end requirements, but rather incremental budgets to be fulfilled within each intermediate AS on an end-to-end session traversing many Autonomous Systems.

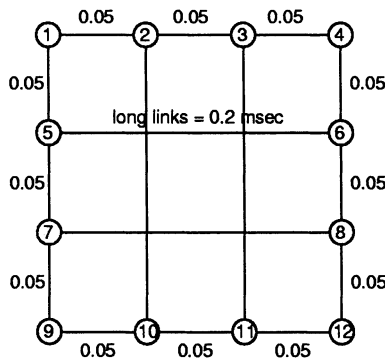
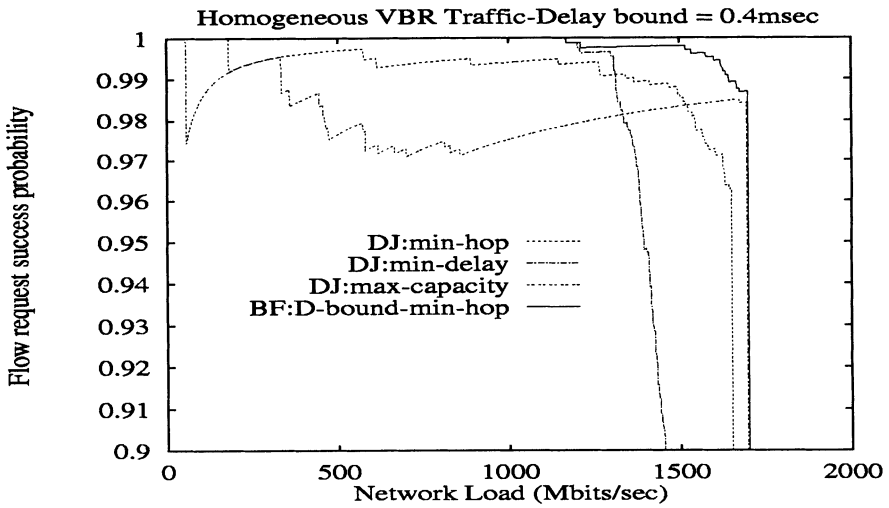


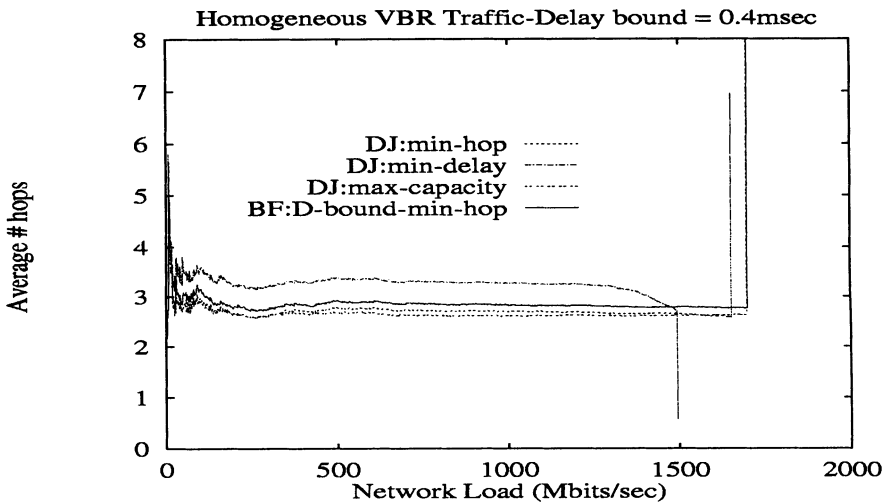
Figure 9 Small Autonomous System

We assume a small network of 12 nodes, depicted in Figure 9. The network has a very simple topology, so that we can force the various routing algorithms to compute different routes. The network diameter is four. Links have 155Mbps speed and propagation delays are as labelled (msec) in the figure. Propagation delays were chosen to reflect a 5Km island. No queueing nor scheduling delays are explicitly considered in the simulations, as they could be easily added to the propagation delays. Buffers are provided on per output basis. A flooding scheme is used to convey link state information across the network. Link state messages are generated periodically, approximately at every 50msec. At each network node, new reservation requests arrive according to a Poisson process, with an average rate of 200 requests per second per node. Once resources are reserved within the network, they are held for the entire duration of the simulation experiment. Thus, eventually the network becomes saturated. The figure of merit is the number of requests which are accepted, that is, the connection success rate.

In the first simulation study, we consider delay bounds only. The sessions require bandwidth  $\geq 1.5$  Mbps, and delay  $\leq 0.4$  msec. Four routing strategies are considered: 1) Dijkstra minimum hop algorithm: a minimum hop path is first generated. If the path satisfies the delay bound, a reservation message is sent through the new path, otherwise, the session request is rejected; 2) Dijkstra minimum delay algorithm: a minimum delay path is computed. The same check on delay bounds is performed; 3) Dijkstra widest capacity algorithm: the path with maximum available capacity is computed. The resulting path is checked against the delay bound, as before; 4) Bellman-Ford delay-constrained minimum hop algorithm: a minimum hop path among all paths within the delay budget is computed. All routing strategies eliminate at each session request the links with insufficient bandwidth. The results are shown in Figure 10.



a) Request success rate



b) Path average # of hops

**Figure 10** Performance of various QoS routing strategies for delay sensitive flows

The Dijkstra minimum hop strategy is quite successful in loading the network up to a high saturation point. Some requests, however, are rejected even

for light network loads. This is mainly because the min-hop strategy is not QoS aware; some of the paths generated by min-hop routing violate the delay bound. For instance, a path request (9,4) will likely generate the path (9,7,5,6,4), which has delay =  $0.45msec > 0.4msec$ , whereas the longer hop path (9,7,5,1,2,3,4) has delay = 0.3, thus within the delay bound.

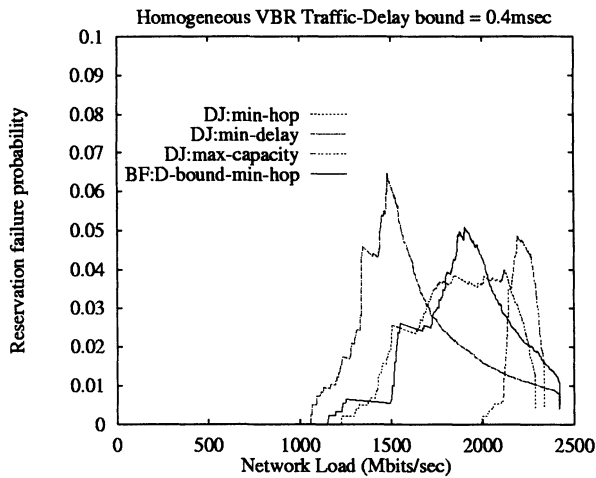
A Dijkstra minimum delay strategy, on the other hand, provides a very high session acceptance rate for light to medium network load states, since if a path is found, this path is guaranteed to comply with the delay bound. However, as some paths are unnecessarily long in hop count (high average hop count, see Fig. 10 b)), the load saturation point is the worst among all routing strategies surveyed. For instance, a request (1,4) will likely result in the path (1,2,3,4,6), with delay = 0.2 msec, whereas the less wasteful path (in network resources) (1,5,6) is available.

Interestingly enough, the Dijkstra widest capacity strategy leads to the best network utilization among all routing strategies. However, it also has the worst performance in success rate, because as the network load increases, uncongested paths quickly become too long to support the delay bound required. For instance, a request (10,9) will likely generate path (10,2,1,5,7,9), which violates the delay bound required, if the delay compliant path (10,9) has less but sufficient bandwidth available to carry the session.

Lastly, Bellman-Ford delay bound minimum hop strategy has the best performance both regarding session request success rate and maximum network load. This is because it selects paths with minimum number of hops within the delay budget.

The average number of hops depicted in Fig. 10 b) agrees with our observations. Notice that, when the network is heavily saturated, all but the minimum delay routing algorithm manage to still find very long (in hop count) suitable paths. That is because the minimum delay routing strategy has used all long paths before the network got into a heavy load state.

The Bellman-Ford QoS aware routing algorithm tries to choose the minimum hop path, among those which satisfy the QoS constraints. We believe that this strategy is more prone to errors than others if the algorithm uses stale information, due to network latency. We have confirmed this conjecture by repeating the simulation above, with a broadcast frequency 100 times slower than before. We have tracked the number of paths believed to be QoS compliant, but which failed to get the resources reserved, due to inaccurate link state information. Since the load/request success rate profile remains unaltered, we report only the percentage of unsuccessful requests due to stale link information (Fig. 11).

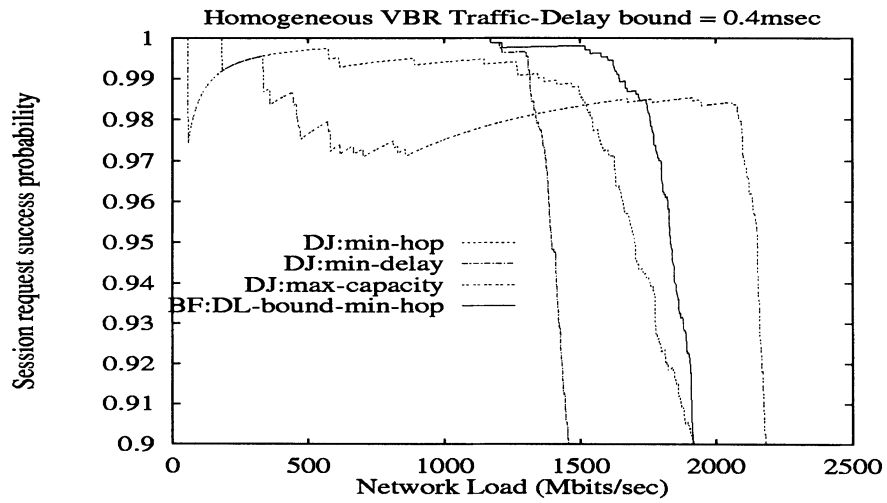


**Figure 11** Impact of stale link state information on the various routing strategies

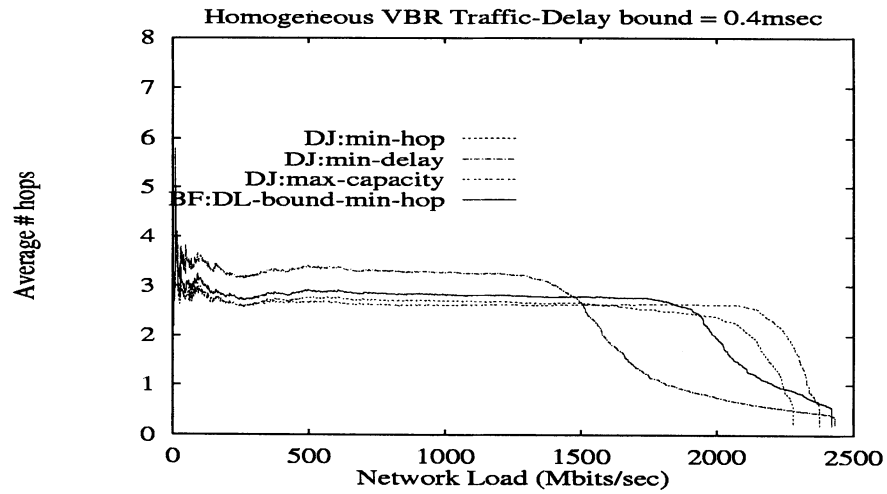
In the next experiment, we have included an additional constraint in our problem, namely, the loss probability metric. Each router must provide a packet loss guarantee of  $10^{-8}$  for each of its links, and the QoS application requires a loss not superior to  $3 \times 10^{-8}$  in addition to an end to end delay  $\leq 0.4msec$ . Results are depicted in Figure 12.

First note that the network saturation point is higher than previously for all algorithms. This is explained by the fact that the loss probability constraint tends to block long sessions (in hop count), which improves network utilization. The additional observation to be made by inspection of Fig. 12 is that the Dijkstra widest capacity strategy leads to the best network utilization (at saturation) among all routing strategies, even better than the Bellman-Ford QoS routing strategy. However, max-cap gives an unacceptably high rejection rate in the non-saturated region. We argue that this is because again the loss probability constraint limits the feasible sessions to a very small number of hops. Within this scenario, the Dijkstra widest capacity routing strategy is successful in routing only very short (in hop count) sessions. Since the QoS Bellman-Ford is able to establish longer sessions, the maximum network load is expected to be smaller than the Dijkstra widest capacity. This can be confirmed by the average number of hop curve displayed by Fig. 12 b). Notice also that the high average hop count of Dijkstra minimum delay strategy is again wasteful, leading to the worst network saturation point. Finally, since long paths (in hop count) are no longer suitable due to the loss probability constraint, no spike is present at the left side of the average hop count curves, as it was the case with the previous simulation experiment.





a) Request success rate



b) Path average # of hops

**Figure 12** Performace of various QoS routing strategies for delay and loss sensitive sessions

## 6 CONCLUSIONS AND FUTURE WORK

We have studied a class of QoS routing which can be solved with a Bellman-Ford algorithm, when the main objective function is the number of hops. We have shown that, for delay jitter and loss probability metrics with certain properties (e.g., constant among links of an AS), a BF algorithm is capable of solving a minimum hop, delay, delay jitter, loss, and bandwidth constrained routing problem. The class of multiple metrics considered in this document is still a bit restrictive. (Henig 1994), among others, has shown that a large class of multiattribute shortest path problems can be solved efficiently. The question whether this large class includes any routing problem of interest for QoS applications in multimedia high speed packet networks is an important research topic.

We have compared a QoS Bellman-Ford algorithm with a widest shortest path, and a pure minimum single metric strategy, including delay and hop count. We have exemplified tradeoffs in using these various routing strategies in reserving resources within an Autonomous System.

Although we have considered a centralized version of the Bellman-Ford algorithm, a distributed version is also possible, if QoS routing research moves towards distributed algorithms. However, we have detected a recent trend towards centralized algorithms, such as QOSPF, rather than distributed ones. Since centralized algorithms require large databases for topology information, QoS hierarchical routing, where routing information is aggregated, is an important research direction. Therefore, we are currently extending the present study to QoS routing across ASs, in a hierarchical routing fashion. The present work points to interesting directions regarding the budget of end-to-end QoS bounds among the various Autonomous Systems. We are also studying how aggregation of resource information may affect the routing algorithms studied in this document.

## REFERENCES

- D. Bertsekas and R. Gallager (1992) *Data Networks*, *Prentice-Hall*.
- Cormen, Leiserson and Rivest (1990) *Introduction to Algorithms*, *McGraw-Hill*.
- M. R. Garey and D. S. Johnson (1979) *Computers and Intractability*, *Freeman*, San Francisco.
- L. Georgiadis, R. Guerin, V. Peris (1996) Efficient Network QoS Provisioning Based on per Node Traffic Shaping, *In Proceedings of IEEE INFOCOM96*, vol.1, pp. 102-110.
- R. Guerin, A. Orda, and D. Williams (1997a) QoS Routing Mechanisms and OSPF Extensions, *In Proceedings of IEEE GLOBECOM97*, Vol. 3, pp. 1903-1908, Phoenix, Arizona.
- R. Guerin, S. Kamat, and S. Herzog (1997b) QoS Path Management with

- RSVP, *In Proceedings of IEEE GLOBECOM97*, Vol. 3, pp. 1914-1918, Phoenix, Arizona.
- M. Henig (1994) Efficient Interactive Method for a Class of Multiattribute Shortest Path Problems, *In Proceedings of Management Science*, Vol. 40, # 7, pp. 891-897.
- K. Lee, K. Kim, H. Choi, K. Kim, and S. Kim (1997) QoS Based Routing for Integrated Multimedia Services, *In Proceedings of GLOBECOM97*, Vol.2, pp. 1047-1051, Phoenix, Arizona.
- C. Parnavalai, G. Chakraborty, and N. Shiratori (1997) QoS Based Routing Algorithm in Integrated Services Packet Networks, *In Proceedings of International Conference on Network Protocols*, Atlanta, Georgia, pp. 167-175.
- Z. Wang and J. Crowcroft (1995) Bandwidth-Delay Based Routing Algorithms, *In Proceedings of GLOBECOM95*, Vol.3, pp. 2129-2133.
- H. Zhang (1995) Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks, *In Proceedings of the IEEE*, 83(10), pp. 1374-1396.
- W. Zhao and S. K. Tripathi (1997) Routing Guaranteed Quality of Service Connections in Integrated Services Packet Networks, *In Proceedings of International Conference on Network Protocols*, Atlanta, Georgia, pp. 175-182.

## 7 ACKNOWLEDGEMENTS

This work has been partially supported by CNPq under Grant 201597/93-4(NV) and by GTE.

## 8 BIOGRAPHY

Dirceu Cavendish was born in Recife, Brazil. He received his bachelor degree in Electronics from Federal University of Pernambuco, Brazil in 1986. He spent five years working with telecommunications in the Business Communications Division of Philips, as a Development Engineer. He received his M. S. in Computer Science from Kyushu Institute of Technology, Japan, in 1994. He is currently a Ph. D. candidate at Computer Science Department-UCLA, working with congestion control and routing in Quality of Service supportive high speed networks.

Mario Gerla was born in Milan, Italy. He received a graduate degree in engineering from the Politecnico di Milano, in 1966, and the M.S. and Ph.D. degrees in engineering from UCLA in 1970 and 1973, respectively. He joined the Faculty of the UCLA Computer Science Department in 1977. His research interests cover the performance systems and high speed computer networks (B-ISDN and Optical Networks).

# Feedback controlled scheduling for QoS in communication systems

*J. Schiller*

*Institute of Telematics, University of Karlsruhe*

*Karlsruhe, Germany*

*j.schiller@ieee.org*

## **Abstract**

Many traditional mechanisms for QoS (Quality of Service) provisioning failed due to complexity, incompatibility to existing applications, or the need for new computer architectures. To be successful, an approach should be integrated into commercial widespread systems, simple, and controllable. This paper discusses basic problems of traditional approaches and proposes a communication system based on feedback controlled schedulers. This enables automatic adaptation to system parameters and changes in the environment. One crucial component for QoS in communication, a scheduler for data packets, will be discussed in more detail. The performance evaluation shows, that it is feasible with the proposed software approach to reach high throughput and to guarantee a broad range of QoS parameters for individual traffic flows.

## **Keywords**

QoS, shaping, scheduling, network adapter, feedback mechanisms, adaptive applications

## **1 INTRODUCTION**

One basic question, that motivated many research activities, is the provisioning of (guaranteed or statistical) QoS for communication oriented applications by the underlying communication subsystem (Campbell, 1996), (Aurrecoechea, 1998). The motivation for QoS is based on the demands of multiple multimedia applications communicating over a large range of communication media (radio,

fibre, copper) within heterogeneous networks (e.g., IP, ATM). Media and networks both provide different QoS to the communication subsystem within a computer, QoS-architectures try to cover these differences and offer an abstract interface with certain parameters (e.g., bandwidth, loss-ratio, jitter, delay).

Many QoS-architectures concentrate on how to provide QoS-parameters to applications and how to map different parameters onto each other. Examples are the mapping of application specific parameters, such as frame rates for a video application, onto system specific parameters like CPU time or memory consumption. These parameters are again mapped onto network specific parameters like maximum bandwidth or average delay. Many of these efforts resulted in a large set of rules and functions for mapping parameters depending on system properties, protocols, and applications (e.g., Campbell, 1994).

However, almost none of these advanced systems is in widespread use due to several reasons:

- History tells, that quite often the simple systems survive, although they are not always better from a technical point of view. Examples are Ethernet and IP, both simple “best-effort” technologies, but still fulfilling their main purpose. The add-on offered by technologies like TokenRing or ATM is not always worth the higher complexity. Although ATM as a more advanced technology offers guaranteed QoS, QoS in IP based networks seems to be the way many people go. Additionally, market forces often favour simplicity (e.g., Cisco, 1997).
- Many of the new QoS architectures also require new operating systems, new applications or even new hardware to show their potential. Generally, users do not switch to complete new systems besides for special applications (e.g., high quality TV studios). The biggest problem for new architectures is the lack of standard applications, such as word processing, spreadsheets etc. Users do not want to have separate computer systems.
- A major problem from a research point of view is the controllability of complex systems. Today’s computer systems do not consist of a single CPU, a single physical memory and execute only one process at a single point in time. Even the simplest PC comprises many processors each with a separate memory. Examples are network adapter, frame-grabber, video adapter, SCSI adapter etc. Many QoS architectures cover only the main CPU, the scheduler of the operating system and the communication oriented applications. Interrupts by network events are hard to control, copying between bus-master adapters cannot be predicted, non-communication oriented programs, such as, e.g., a compiler, are neglected. A manageable QoS architecture can, thus, not cover all system parameters, but still has to try to supervise the system precisely and to apply the correct mapping functions. Thus, these approaches often try to apply exact rules on fuzzy parameters to get exact results. Quite often unexpected system events render these decisions useless.

This paper proposes a system architecture with the following features that incorporates fuzziness of system parameters, reacts on changes and still tries to give certain guarantees for communication applications:

- The architecture is simple and does not require a large set of system parameters.
- Existing applications still work, the proposed system enhancements can be integrated into COTS (commercial off the shelf) operating systems (e.g., WindowsNT, WindowsCE, Unix) and standard computer architectures (PCs).
- Users must be “relieved” from complicated new features, applications have to adapt autonomously to changes, users can utilise the enhancements via a very simple interface.

The remainder of the paper is organised as follows. The next section discusses QoS mechanisms in more detail and presents a general system architecture based on two internal feedback loops which can be integrated into common systems and incorporates the idea of fuzzy parameters. The third chapter shows more detailed one main component within the architecture, a packet scheduler for QoS provisioning in the lower communication layers. Finally, the conclusion outlines key features and the next steps towards a complete implementation of the proposed architecture.

## 2 SYSTEM ARCHITECTURE FOR QOS VIA FEEDBACK CONTROLLED SCHEDULERS

Two fundamental approaches dominate solutions for QoS, one is QoS management architectures, the other comprises adaptive applications. While the first typically requires exact knowledge about the operating system, computer architecture etc., the latter assumes the ability of adaptation to changes in resources from applications and protocols.

- *QoS management*: these approaches comprise several mechanisms, such as resource reservation in advance, mapping of QoS parameters, admission control, QoS (re)negotiation etc. If these mechanisms are applied and all resources within systems and networks are considered, this is a way to guarantee QoS from application to application on end-systems. However, several factors make the realisation of these approaches extremely complex and impractical. Due to the worst case reservations needed for hard QoS guarantees, resource usage is typically very inefficient. Furthermore, it is well known that it is in general not possible to predict the resource requirements for multimedia applications. Additionally, many QoS management architectures require changes to user interfaces and applications.
- *Adaptive applications*: this approach, typically, does not give hard guarantees for QoS. Every application starts with an initial amount of resources and tries to provide the best QoS possible based on the actual available resources (Gecsei, 1997). It is quite obvious that this approach cannot offer constant or predictable QoS for users.

### 2.1 Feedback mechanisms for QoS provisioning

To circumvent the disadvantages of traditional QoS management frameworks, newer approaches suggest simple feedback mechanisms and a set of adaptive resource managers (Diot, 1997) (Lakshman, 1997). The basic idea is to use

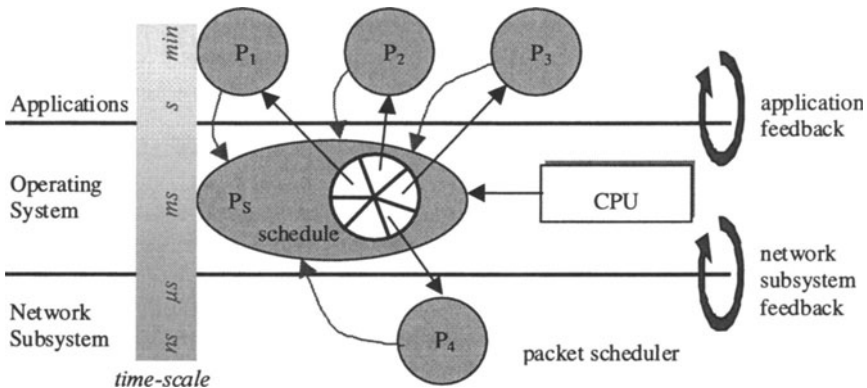
feedback mechanisms within the operating system to provide QoS without exact knowledge about the actual system load or the underlying hardware. Every application providing QoS compares its own requirements continuously with the real offered QoS and generates a feedback signal out of the difference. Based on this signal the operating system redistributes the resources assigned to applications.

These approaches do neither require pre-calculated resource requirements for applications nor do they need complex QoS (re)negotiation. This avoids the disadvantages of complex frameworks and does not necessarily require changes to existing applications. Clearly, these approaches cannot give hard guarantees. In overload situations a decrease in QoS will be experienced, which is inevitable in systems without hard resource reservations. Furthermore, these end-system centric approaches cannot control the whole distributed system with feedback loops between different applications on different end-systems.

However, for many users this “soft” QoS on their own end-system is satisfactory. Users will notice at once performance degradation caused by themselves and can then decide how to react, e.g., stop the new application, accept the degradation, or stop another application. Co-existence of hard and soft QoS mechanisms is possible, but out of scope of this paper. Controlling the whole distributed system requires similar mechanisms on all end-systems participating and appropriate control loops in-between. This does not seem to be realistic as a first step in a world of highly heterogeneous systems, networks, and applications.

## 2.2 Layered feedback loops

A closer look into the system architecture of end-systems shows, that the implementation of a single feedback loop is not useful due to the very different time scales. For example, the scheduler and traffic shaper presented in chapter 3 works within microseconds, CPU-schedulers are typically based on milliseconds while interactions with users happen in the second to minute range. Thus, this work identifies three different layers within a system, separated by the different timing requirements for changes within the layer (c.f. figure 1). The layers are then coupled via feedback loops as explained in the following.



**Figure 1** Layered feedback loops.

The three layers are the application layer with application processes, the operating system layer with the scheduler and the network layer with the packet scheduler and integrated traffic shaper. This architecture decouples time-critical events in the network layer from longer term changes in the application layer. Management of the CPU as the central resource is done via the scheduler which is in-between the two feedback loops.

While chapter 3 focuses on the lowest layer and presents an already fully implemented software solution to provide QoS, the following gives some details regarding the feedback loops in general.

### *Network subsystem feedback*

The packet scheduler of the network subsystem determines which packet, or in the case of ATM which cell, will be sent at what point of time. A software based approach as presented in chapter 3 will need a certain number of CPU cycles to perform this task. If the scheduler cannot get the required cycles, packets will be delayed. Based on this delay a feedback signal is generated and returned via a (software) low-pass filter to the CPU scheduler. The CPU scheduler can now assign more CPU time to the packet scheduler, enabling it to perform the scheduling closer to an optimum. This feedback can be extremely fast and without any user intervention. Additional overhead for determining actual system load is not necessary.

### *Application feedback*

In a similar manner a feedback is constructed between applications and the CPU scheduler. Now two inputs are possible. One input comes from the application itself noting that it needs more cycles. This can be done via measuring, e.g., playout buffers. This mechanism is necessary to maintain a certain level of quality without any user intervention. The second input comes from an unsatisfied user. A system offering users too many controls and parameters to change the system behaviour will not be accepted. However, integrating a simple “Quality” button for the user to express his or her dissatisfaction is an easy to understand interface (c.f. figure 2). If the user hits this button, the CPU scheduler tries to assign more CPU cycles to the appropriate application and the processes on which this application depends. Pressing this button while holding the “shift”-key frees CPU cycles from this application, thus, potentially rising the performance of the other applications. This is needed if a user is dissatisfied with the performance of the other applications controlled by the scheme. Using more complex controls, e.g., sliders, requires a clear semantics behind shifting the slider. It is not at all obvious for a user to understand the implications of shifting a slider half-way up or 10% down for a video or audio application.



**Figure 2** Simple user interface via a Quality button.



The application shown in figure 2 is up to now a very simple test module which can generate a signal based on the **Quality** button to increase/decrease the CPU time associated to the process while copying data to/from another end-system. For this first step only standard operating system calls of WindowsNT have been used to influence the scheduler for it is not possible to change the built-in scheduling mechanisms themselves.

It is obvious, that these mechanisms always try to steal cycles from other applications in case of overload situations to favour an application the user determines. Assuming finite resources and no detailed knowledge about the underlying system this is the best one can do without complicated user interaction.

The big advantage of this "soft" QoS solution is its applicability to any system, from a low performance PDA with WindowsCE to a standard PC running UNIX or WindowsNT without changes to standard applications or the operating system. Furthermore, it will be easily accepted also from non-technical users who do not understand the implications of, e.g., changing resolution or colour depth of a video conferencing tool, switching between different audio encodings etc.. Adaptive applications necessary for this approach are already available, examples are vic (Busse, 1996), vat (Jacobson, 1995), IVS (Bolot, 1994) etc.

### 3 SOFTWARE-BASED PACKET SCHEDULING WITH ADVANCED SHAPING

While up to now a general architecture for QoS provisioning was discussed, the following presents detailed mechanisms for the packet scheduler residing in the lower layer of figure 1. This process is crucial for the provision of QoS and is typically implemented in hardware for ATM interfaces (so called traffic shaper chips) or missing in the case of best effort adapters (e.g., Ethernet).

Today's ATM networks and the future Internet both support QoS. Many discussions are going on how to provide what kind of QoS. However, it is quite clear, that basic support of packet scheduling and traffic shaping is necessary. While many adapters for the Ethernet simply push packets ready to send onto the network and, thereby, create very bursty traffic, enhanced systems have to create a certain shape of traffic to stay within limits of some traffic contract. The following discussion is based on the ATM terminology, because up to now it is not so obvious what parameters will be defined within an enhanced Internet. However, this paper shows, that scheduling and shaping is feasible for ATM for a high bitrate, thus, one can conclude that this will be even easier for the much longer IP packets compared to the short ATM cells.

#### 3.1 Existing hardware shapers

Existing shaping solutions are typically based on dedicated hardware chips. The scalability of all chip-based solutions is very limited. Values like the maximum number of supported connections (VC) are of more theoretical nature and typically only limited by the width of registers. More interesting is the number of *simultaneously* supported connections or the number of connections supported *on-chip*. These numbers of supported connections rely not only on the amount of

memory to store connection state data, but rather on the algorithms; data structures, and available processing power for traversing connection information.

Furthermore, there are typically limits on the maximum number of *different* traffic characteristics. Additionally, the question arises what parameters can be set and if they can be used independently. Typical restrictions are 8-12 PCR (Peak Cell Rate) queues, every connection has to be in one of these queues (Fujitsu, 1997), (SIEMENS, 1997). Furthermore, the SCR (Sustainable Cell Rate) is often derived from the PCR via a per connection ratio (e.g.,  $\frac{1}{2}$  or  $\frac{1}{4}$  of the PCR). These restrictions are due to the fact, that the cell rates are generated using explicit hardware counters, only a very limited number of these fit on a chip (c.f. (ATM Forum, 1996) for further explanation of ATM traffic parameters).

The isolation between different connections sharing one PCR queue is typically not addressed in any of the evaluated solutions. Some solutions provide additionally priority classes, the behavior within a priority class during transient overloads is not further determined.

It is important to compare the capabilities of the UPC (Usage Parameter Control) chips located at the UNI (User Network Interface) with those of the shapers due to the fact that the UPC chips will decide if an incoming cell of a connection shaped by one of the shapers is accepted or not. The first fact one notices is the more detailed control capabilities and the variety of parameters to check. For example, the ATM\_POL3 from ATecoM (Atecom, 1997) can control up to 64k VCs checking PCR, SCR, CDV (Cell Delay Variation) and maximum burst size using a dual leaky bucket per VC. IgT has the WAC-186-B (Integrated, 1997) which uses GCRA (Generic Cell Rate Algorithm) to monitor PCR, SCR, CDV and burst tolerance for up to 16k active VCs for a bitrate up to 250 Mbit/s. Finally, the BNP2010 UPC of National Semiconductor (National, 1997) checks up to 16k VC with 3 GCRA's per VC up to a speed of 622 Mbit/s. A simple comparison shows, that the UPC can be much more precise and, hence, more restrictive than the capabilities of the best shaper chips. Altogether, this discrepancy may have the effect of cell-loss at the UNI although sender and UPC agreed upon the same traffic parameters in the traffic contract.

### **3.2 Software process for shaping and scheduling**

The CPU has to run a process for packet scheduling and traffic shaping. This process makes sure that the right cell will be sent at the right time according to the traffic contract. The shaper function manipulates context information for all connections, i.e., Peak Cell Rate, Sustainable Cell Rate, Minimum Cell Rate, number of PDUs etc. This function also provides the scheduler function with the necessary timing information for the next cell of the current connection, i.e., the earliest point in time this next cell can be sent without violating the traffic contract. Using this information the scheduler function updates its sorted list of connection identifiers (realized as a heap without pointer operations). The criterion for sorting is the time the next cell of a connection can be sent. The scheduler function returns a pointer to the context data for the connection on top of the sorted list. This selection of the next eligible cell has to be done within, e.g., 2.8 $\mu$ s for a 155Mbit/s

ATM adapter if there were no other processes running. For more technical details see (Schiller, 1998).

The shaper function is based on a combination of the VSA (Virtual Scheduling Algorithm) and LBA (Leaky Bucket Algorithm), both proposed as possible algorithms for the GCRA at the UNI to a public ATM network (ATM Forum, 1996). The implementation presented uses these algorithms for actively creating (and thereby shaping) traffic instead of controlling traffic as UPC at the UNI. If the parameter for delay variation CDVT (Cell Delay Variation Tolerance) is set properly, this approach guarantees the acceptance of cells at the UNI.

### 3.3 Priorities and Proportional Sharing

A very important question is how a scheduler behaves in overload situations. Overload situations can occur quite frequently, if one does not want to make only conservative reservations, i.e., allowing the sum of all PCRs never to be greater than the total capacity of the link. This would result in a very poor overall utilization if, e.g., VBR (variable bit-rate) traffic sources are used. Here the PCR can be easily 10-1000 times larger than the SCR. One example is the transfer of MPEG2 coded video streams. Typical values are 1.0 to 15.0 Mbit/s for PCR, 0.2 to 4.0 Mbit/s for SCR. In addition, quite often a priority scheme is required to weight different traffic streams. One could for example give voice connections a higher priority compared to connections fetching pictures from Web-pages. This would result in a higher audio quality and only minimal additional delay for the picture data transfer.

Fairness in overload situations is an important feature of a scheduler. If a user has started, e.g., several video applications that load the network completely and now starts an additional one he or she expects the available bandwidth to be shared fairly between the applications. The communication system can not make any assumptions of the importance of an application, and, therefore, a proportional sharing scheme is the best one can do. That means, that an application that used twice the bandwidth compared to another one still gets twice as much as the other one. But now this is less than before due to the overload. Thus, the implementation fulfills the criterion of ideal fairness as defined in (Varma, 1997). To privilege one application, one can shift the application to a higher priority. The implementation guarantees this proportional share within one priority class independently for every connection. The settings of the proportional sharing between different priority classes is left to the operator and depends on the policy of a service provider.

Prioritizing an application leads to questions concerning the interference between priority classes. Depending on the scheduling policy one can decide for a hard priority scheme, i.e., the scheduler tries first to satisfy connections with higher priorities and ignores lower priorities in case of an overload. This results in starvation of connections in lower priority classes. An alternative solution could provide a minimum share of the total bandwidth to avoid starvation. This is in general the better alternative due to the fact that overload situations are typically transient and common communication protocols like, e.g., TCP cannot deal properly with a total starvation but adapt well to lower bandwidth. The

implementation allows both alternatives by guaranteeing proportions of the total bandwidth in an overload situation.

Proportional sharing and the overload behavior described above are up to now not implemented in any of the available traffic shaper chips. Either these implementations avoid overload situations by not allowing over allocation (LSI, 1997) or they throttle the total traffic via a leaky bucket (Fujitsu, 1997) (SIEMENS, 1997).

### 3.4 Best effort services

The scheduler implemented supports not only CBR (constant bit-rate) or VBR in an ATM context. Via the priority and sharing mechanisms it is also possible to support best effort services like UBR (Unspecified Bit Rate) and adaptive services like ABR (Available Bit Rate), (ATM Forum, 1996). UBR traffic is given a PCR rate, which could be below the actual link rate to avoid unnecessary cell loss downstream. If there is UBR traffic to send, it will use idle cell slots, but back-off as soon as other traffic sources have data to send or its PCR limit is reached. UBR traffic can be handled by assigning the lowest priority to this traffic class. The PCR could be dynamically set.

ABR does not need a new mechanism, since it can be seen as VBR with dynamically adjusted rate. In ABR, a minimum user requested rate is guaranteed. As opposed to UBR, ABR should get a fair share even in transient overload situations. Thus, ABR traffic is assigned to a higher priority class than UBR. The ABR feedback mechanism is independent of this work and the PCR is set dynamically in the connection states according to the feedback channel. The shaper and scheduler will then automatically adjust the rates.

A straightforward priority set-up is to give CBR traffic the highest priority, followed by VBR-rt (real-time), VBR-nrt (non real-time), ABR, and finally UBR with the lowest priority. The algorithm can now guarantee the different classes a minimum share, except for UBR which does not need one. Within priority classes proportional sharing is applied, every priority class can get the full bandwidth if no cells from other connections at higher priority levels are ready for transmission.

### 3.5 Scheduling variable sized packets

The basic scheduling mechanism is also applicable to variable packet sizes. One example of variable packet sizes is the scheduling of IPv6 packets, where the *flow id* is used as a connection identifier. One way to schedule variable sized packets is to change the parameters used in the algorithm as follows. Rates are expressed in bytes/s instead of cells/s, and the earliest sending time is adjusted according to the size of the packet. The tokens in the LBA now count bytes instead of cells, which ensures that a connection gets bandwidth up to its share and also the long term fairness. Variable packet sizes will inevitably introduce delay variance when multiplexed. Once the adapter has started to send a packet it cannot be interrupted and will block other, possibly shorter and more urgent packets. For all networks there exists a maximum packet size, e.g., 1500 bytes for Ethernet, which puts an upper limit on the delay caused by multiplexing and blocking.

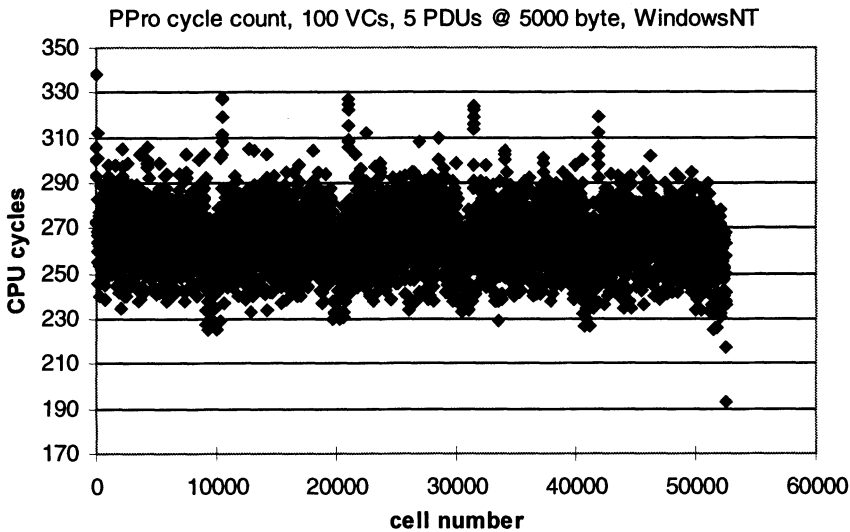
If there are enough tokens in the bucket, the packet is eligible for sending as soon as possible. The scheduler should then find the earliest sending time for the connection which depends on already scheduled packets for other connections. If there are not enough tokens, the scheduler has two alternatives. The first is to postpone the scheduling until there are enough tokens. The other alternative is to calculate when there will be enough tokens and then schedule the packet for that time. The first alternative is not attractive since we need to activate the scheduler again when there are enough tokens available (which is indeed predictable). The second alternative also has a drawback, since it may generate bandwidth internal fragmentation. If a packet is scheduled in the future, it may be the case that earlier empty slots can not be used since they are too small to fit a large packet. This is not the case for ATM cells, since they have fixed sized cells.

Our current algorithm works according to the second principle. A slow connection packet will hence be scheduled far away in the future. There are some unsolved fairness issues here that need to be considered. For example, a slow connection with large packets will have little or no delay variance while a high bandwidth connection may have problems to find a big enough slot and will be delayed. The fairness problem, a detailed evaluation, and performance measurements are part of ongoing work.

### 3.6 Implementation and performance evaluation

The algorithms were implemented on different platforms, such as PentiumPro, Pentium-II, Alpha, UltraSPARC running the appropriate standard operating systems (WindowsNT, Linux, Digital UNIX, Solaris). All measurements show the number of CPU cycles needed for calculating the sending time for the next cell of a connection, resorting the list of active connections, and initiating the actual transfer of the cell. Absolute upper limits are, e.g.,  $2.8\mu\text{s}$  for a 155Mbit/s adapter, 680ns for a 622Mbit/s adapter if no other processes are running. Assuming application processes, this time is certainly reduced. All measurement use worst case scenarios, i.e., the scheduler always had to resort the complete list. Furthermore, running the full operating system guarantees realistic results for the execution times compared to stand alone systems often used for performance figures.

The example in figure 3 shows clearly, that with today's processors this task can be done fast enough and no dedicated scheduling and shaping hardware is needed. In the example 100 connections are scheduled, each connection sends 5 PDUs with 5000 bytes each. Clearly visible in this example are 5 peaks for issuing DMA commands to prefetch new data at the start of a new PDU. For more implementation details see (Schiller, 1998). The measurements took place on a 200MHz PentiumPro running Windows NT 4.0. The average of 270 cycles equals  $1.35\mu\text{s}$ . Assuming a 155 Mbit/s adapter, i.e., generating a cell every  $2.8\mu\text{s}$  in worst case, this example leaves ca. 50% of the CPU power for other processes. Further experiments showed that scheduling and shaping of 64000 individual connections requires 620 cycles on average, 1000 connections require 330 cycles on a PentiumPro architecture. The shaping function is in  $O(1)$ , the scheduler in  $O(\log_2 \# \text{connections})$ .



**Figure 3** Scheduling and shaping of 100 individual active connections.

#### 4 CONCLUSIONS

The paper discussed basic problems of traditional approaches for QoS provisioning, mainly complexity and incompatibility, and proposed a simple architecture based on feedback controlled schedulers. This approach enables automatic adaptation to changes in system parameters and the environment. The main component for QoS in the network subsystem, a packet-level scheduler with advanced shaping, was explained more detailed. The performance evaluation showed, that it is absolutely feasible with the software approach to reach the throughput necessary for high-performance adapters and to guarantee a broad range of QoS parameters for individual traffic flows in case of ATM. In case of future IP-based networks with differentiated services as currently discussed (Wroclawski, 1998) the CPU load caused by scheduling and shaping is even lower. Thus, one next step of our work is the application of the algorithms to variable sized IP packets and standard Ethernet adapters to provide the shaping necessary for differentiated services.

As discussed in chapter 2 the fully implemented software-based scheduler is only one component within the system. The current very basic add-ons to the operating system scheduler have to be extended and refined. In case of Linux this includes more changes to the scheduler itself, in case of WindowsNT a more powerful hierarchical scheduler on top of the standard scheduler. It is obvious that this approach cannot change commercial operating systems themselves, but has to work on top of them to remain compatible. Furthermore, the up to now basic integration of the Quality-button into adaptive applications has to be improved. Another important topic is the controllability of the feedback loops, especially together with external protocol or application dependant loops (TCP, ABR etc.).

Up to now, decisions are only local for we cannot control the whole distributed system. Here the integration of fuzzy control is under investigation to provide stability and integrate imprecise parameter values. While the whole system may be not as powerful as dedicated architectures with complex QoS management, the big advantage is, that it can be integrated into commercial of the shelf systems.

## 5 REFERENCES

- AtecoM (1997) ATM\_POL3, <http://www.atecom.de/>
- ATM Forum (1996) Traffic management specification, version 4.0, ATM Forum
- Aurrecochea, C., Campbell, A., Hauw, L. (1998) A Survey of QoS Architectures, *Multimedia Systems Journal*, Special Issue on QoS Architecture. May 1998
- Bolot, J.-C., Turletti, T. (1994) A rate control mechanism for packet video in the Internet. *IEEE Infocom*, Toronto
- Busse, I., Deffner, H., Schulzrinne, H. (1996) Dynamic QoS Control of Multimedia Applications based on RTP. *Computer Communications*. January 1996
- Campbell, A. (1994) A Quality of Service Architecture, *ACM Computer Communication Review*, April 1994
- Campbell, A., Aurrecochea, C.: Hauw, L. (1996) A Review of QoS Architectures, *Proceedings of the 4<sup>th</sup> International IFIP Workshop on QoS (IWQoS 96)*, Paris
- Cisco (1997) Cisco's Packet over SONET/SDH (POS) Technology Support; Mission Accomplished. Cisco Systems, Inc., White Paper
- Diot, C., Seneviratne, A. (1997) Quality of Service in Heterogeneous Distributed Systems. *Proc. of HICSS'97*, 30. Hawai'ian Int. Conf. on System Sciences
- Fujitsu (1997) ALC (MB86687A), <http://www.fujitsu.com>
- Gecsei, J. (1997) Adaptation in Distributed Multimedia Systems. *IEEE Multimedia*. April/June 1997
- Integrated Telecom Technology (1997) WAC-186-B, <http://www.igt.com>
- Jacobson, V. (1995) Internet Multimedia Systems. Tutorial, ACM SIGCOMM, London
- Lakshman, K., Yavatkar, R., Finkel, R. (1997) Integrated CPU and Network-I/O QoS Management in an Endsystem. *Proc. of 5<sup>th</sup> IFIP International Workshop on Quality of Service (IWQoS 97)*
- Landfeldt, B., Seneviratne, A., Diot, C. (1998) User Services Assistant: An End-to-End Reactive QoS Architecture. *Proc. of 6<sup>th</sup> IFIP International Workshop on Quality of Service (IWQoS 98)*
- LSI Logic (1997) ATMizer II (L64363), <http://www.lsilogic.com>
- National Semiconductor (1997) BNP2010 UPC, <http://www.national.com>
- Schiller, J., Gunningberg, P. (1998) Feasibility of a Software-based ATM cell-level scheduler with advanced shaping. *Broadband Communications'98*, International Federation of Information Processing IFIP, Chapman&Hall Pub.
- SIEMENS (1997) SARE (PBX4110), <http://www.siemens.de>
- Varma, A., Stiliadis, D. (1997) Hardware Implementation of Fair Queuing Algorithms for Asynchronous Transfer Mode Networks. *IEEE Communications Magazine*, 35(12)
- Wroclawski, J. (1998) Differential Services for the Internet. <http://diffserv.lcs.mit.edu/>

# Scheduling algorithms for advance resource reservation

*Charlie Xu and J.W. Wong*

*Department of Computer Science, University of Waterloo  
Waterloo, Ontario, Canada N2L 3G1*

*Telephone: 519-885-1211, Fax: 519-885-1208*

*e-mail: jwwong@bcr.uwaterloo.ca*

## **Abstract**

Advances in high performance networking have provided improved support to resource intensive applications, e.g., video conferencing and video-on-demand. For conferencing applications, the ability to reserve network resources in advance is highly desired. Most previous works on advance resource reservation are based on a single-link model. We extend this model to a network environment with emphasis placed on multi-party conferencing. A reservation request is characterized by its start time, bandwidth requirement, holding time, and participating parties. Both immediate and delayed acknowledgments are considered. In immediate acknowledgment, the response to a reservation request is returned as soon as possible. In delayed acknowledgment, requests are batched for scheduling purposes with a view that the system may be able to make more informed decisions. However, the delay may be unacceptable to the requester. In this paper, scheduling heuristics are developed for delayed acknowledgment. Simulation results on the performance difference of these heuristics and immediate acknowledgment are presented.

## **Keywords**

Resource management, advance reservation, scheduling

## **1 INTRODUCTION**

Traditionally network services, such as file transfer and remote access to databases, generally operate on an “on-demand” principle. This means that the user submits a request for service at the very moment that he/she wishes the communication to take place. Most studies on network resource management are based on this scenario. Advances in high performance networking have provided improved support to resource-intensive applications, e.g., video conferencing and video on demand. These applications invariably require some quality of service (QoS) guarantees from the network to ensure efficient operation. Much research has been done, and active research is underway, on



the various aspects of a communication network that supports QoS guarantees. A key assumption is again the on-demand nature of service requests. However, for conferencing applications, the need for successful call setup at a pre-specified time argues for advance resource reservation (ARR), i.e., resource reservation made some time before the actual setup of a call. The need for this option in high-speed networks is mentioned in CCITT's Recommendation I.121 on broadband aspects of ISDN [1] and in IETF's Internet draft on session initiation protocol [2].

Performance investigation of ARR systems have been reported in [3-10]. These studies are concerned with the design and evaluation of scheduling algorithms for ARR using single link models. In other studies, architectural issues of ARR systems [11] and mechanisms for establishing a reservation [12] have been investigated. In this paper, we extend previous works on single-link models to a network environment. Our focus is on multi-party communication since it is likely to be the main user of ARR systems. Moving to a network environment introduces new research challenges in network resource management such as scheduling and routing of calls reserved in advance.

A number of network resources can be considered as candidates for advance reservation. Examples are bandwidth and buffer space. For simplicity, we only consider bandwidth reservation, which is the dominant factor in QoS guarantees. The amount of bandwidth required is specified in a reservation request. For example, voice traffic would require 64 Kb/s or less, depending on the encoding algorithm used. Video may also be transmitted at a range of data rates, depending on the encoding algorithm and the quality level.

A important consideration in ARR is the acknowledgment delay, which is the elapsed time from when a reservation request is made to when the result of the request is known. From the perspective of the requester, the acknowledgment delay should be kept to a minimum (referred to as *immediate acknowledgment*). However, by batching requests, an ARR system may be able to make more informed scheduling decisions, and utilize network resources more efficiently. Hence, there may be an advantage in delaying scheduling decisions, and thereby delaying the acknowledgments. Of particular interest to this investigation is the performance benefits of *delayed acknowledgment* over immediate acknowledgment.

In general, exact analytic results for the performance of reservation systems are difficult to obtain, even for single-link models. Most investigations rely on the use of simplified models, approximate analysis, or simulation. Network-scale models are even more complex. For example, routing of multi-point connections is known to be NP-hard except for some specific network topologies [13]. Our investigation is based on simulation. We first develop scheduling heuristics for delayed acknowledgment, and then evaluate the performance difference between these heuristics and immediate acknowledgment.

This paper is organized as follows. Section 2 defines the performance model and the performance metrics used in our investigation. Our scheduling heuris-

tics are developed in section 3. In section 4, results on performance comparison of immediate acknowledgment and delayed acknowledgment are presented. Finally, section 5 contains a summary of our findings.

## 2 PERFORMANCE MODEL

In general a complete model of an ARR system contains the following components: the network model, the request characteristics, and the reservation model. These components, as well as the performance metrics, are defined in this section.

### 2.1 Network model

The network under consideration consists of a set of switches connected by communication links and is modeled by a connected graph  $G = (V, E, w)$ , where  $V$  is a set of nodes representing the switches,  $E$  is a set of edges representing the communication links, and  $w : E \rightarrow R$  is a link cost function. It is assumed that the capacity, or bandwidth, of a link is organized into  $N$  fixed-sized bandwidth units. For example, with 1 Kb/s units in a 1.5 Mbps channel,  $N$  is 1500. A request may only reserve a multiple of these units. On a given link, the difference between the total capacity and the reserved capacity is the amount of capacity still available for reservation or other services. This amount of capacity is called the *residual capacity*.

In our simulation experiments, we use random graphs with 100 nodes as our sample networks. The average degree of a node (number of links adjacent to a node) is 3. The random graphs are generated using the Stanford GraphBase package [14].

### 2.2 Request characteristics

In general, each party in a multi-party conference may need different amount of bandwidth for sending and receiving data. For example, a party has one outgoing video/audio stream but may have to receive several incoming streams. This asymmetry may result in a network with asymmetric residual capacities on the links. Routing a multi-party connection with asymmetric capacity requirements is a very hard problem. A simplification to this problem is the routing of a point-to-multipoint connection where one of the parties is the source and the other parties just listen. This is the Steiner tree problem in directed networks (STDN). Good approximation techniques are not available for this problem. Recently two groups of authors obtained non-trivial approximation algorithms for STDN [15, 16]. Their algorithms achieve an approxi-

mation ratio of  $i(i-1)k^{1/i}$  in time  $O(n^i k^{2i})$  for any fixed  $i > 1$ , where  $k$  is the number of parties. For  $i = \log k$ , they obtain  $O(\log^2 k)$  approximation ratio in quasi-polynomial time.

To reduce complexity, we assume that for a given reservation, the capacity requirement on both directions of each link is the same. We represent a call request by a 4-tuple  $(c, s, h, p)$ , where  $c$  is the capacity requirement;  $s$  is the start time;  $h$  is the holding time; and  $p \subseteq V$  is the set of parties. Each request, if accepted, will result in reservations along a tree spanning all the parties. The same capacity  $c$  is reserved in both directions on each link of this tree. This is called the *uniform capacity assumption*.

For convenience, we assume, as in [3], that time on each link is divided into fixed length units called *slots*. At most one request may arrive in a slot and such an arrival occurs with probability  $\lambda$ . The start time of a request can only be at the beginning of a slot and the holding time is an integral multiple of slots.

### 2.3 Reservation model

The timing diagram in Figure 1 shows how a call request is handled [3, 11]. This request is submitted at time  $t_{\text{req}}$ , and an acknowledgment is returned at time  $t_{\text{ack}}$ . At the requested start time  $t_{\text{begin}}$ , the reserved resources are allocated and the conference begins. The difference between  $t_{\text{req}}$  and  $t_{\text{begin}}$  is called the *notice interval*.

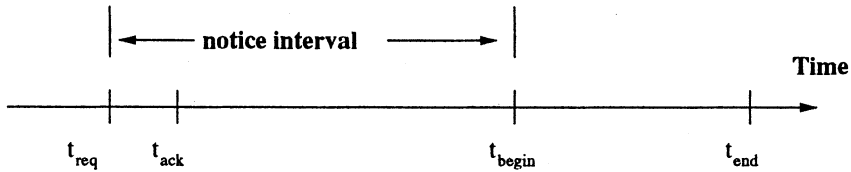


Figure 1 Timing diagram for request handling

Because the available bandwidth is finite, it is not always possible to accommodate all call requests. One approach to handling conflicts is to find an alternative start time, preferably close to the requested start time. Another approach is to reject any requests that cannot be accommodated. It is up to the requester to submit another request for an alternative start time. This is often referred to as a *loss* system. This paper is concerned with loss systems only.

As mentioned in the introduction section, reservation requests can be handled using immediate acknowledgment or delayed acknowledgment. Modeling of immediate acknowledgment is straight-forward. Each time a request is submitted, a scheduling decision is made immediately.

For delayed acknowledgment, the model in [3] is adopted (see Figure 2). Time on each link is partitioned into *reservation periods* of length  $Y$  slots. A *decision point* occurs at the start of each reservation period. When a decision point is reached, all requests associated with that decision point are scheduled. For a given request, its associated decision point is characterized by a global parameter  $P$ , called the *decision point offset*. Consider the arrival of request  $k$ . The earliest possible decision point  $E_k$  is the first decision point after the arrival. The latest possible decision point  $L_k$  is the latest decision point before the requested start time. In this model, the decision point for request  $k$  is given by  $\min(L_k, E_k + P)$ . When  $P = 0$ , the system behavior is similar to that of the immediate acknowledgment model. Increasing  $P$  has the effect of delaying the scheduling decision. If  $P$  is so large that all the requests are scheduled at their latest possible decision point, the behavior is similar to that of a system where batching of reservation requests is maximized.

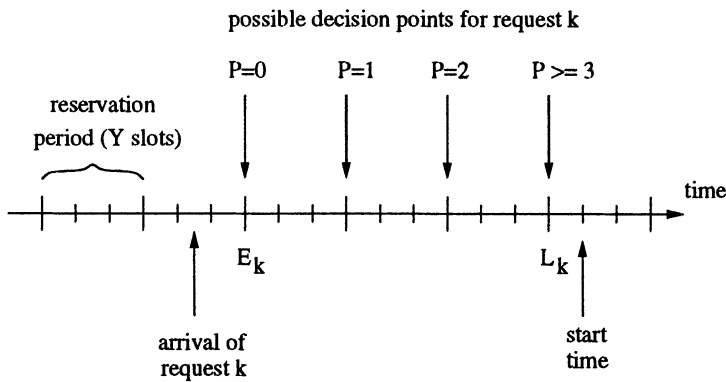


Figure 2 The delayed acknowledgment model

## 2.4 Performance metrics

In previous works on single-link models, performance metrics such as blocking probability and channel utilization were used. The extension of ARR to a network environment introduces the need for additional performance metrics.

In this study, we consider blocking probability and two other performance metrics. The first is related to load balancing. It may be desired that the reserved capacity on the links be balanced so as to better serve traffic classes other than advance reservation, e.g., on-demand requests. The variance of residual capacities is used as our performance metric. Suppose the residual capacity of a link  $e$  at time  $t$  is  $r_e(t)$ , the variance of residual capacities of the

network  $G$  at time  $t$  is defined as

$$\sigma_G^2(t) = \frac{\sum_{e \in E} (r_e(t) - \bar{r}_G(t))^2}{|E|} \quad (1)$$

where  $|E|$  is the number of links and  $\bar{r}_G(t)$  is the average residual capacity of the network at time  $t$  over all the links:

$$\bar{r}_G(t) = \left( \sum_{e \in E} r_e(t) \right) / |E|$$

Over a period of time from  $t_0$  to  $t_1$ ,  $t_1 > t_0$ , we define the time-averaged variance of residual capacities as

$$\bar{\sigma}_G^2(t_0, t_1) = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} \sigma_G^2(t) dt \quad (2)$$

The second performance metric is the total reserved capacity. It is defined as the sum of reserved bandwidth on each link in the network. Let  $b_e(t)$  be the amount of the bandwidth reserved at link  $e$ , the total reserved capacity in  $G$  at time  $t$  is

$$C_G(t) = \sum_{e \in E} b_e(t) \quad (3)$$

The time-averaged total reserved capacity in network  $G$  from time  $t_0$  to  $t_1$ ,  $t_1 > t_0$ , is

$$\bar{C}_G(t_0, t_1) = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} C_G(t) dt \quad (4)$$

These above performance metrics implicitly assume that all links are equal in their importance. In our simulations, we experiment with several networks and average the results. We will therefore drop the subscript  $G$  in our discussions.

### 3 SCHEDULING HEURISTICS

#### 3.1 The scheduling problem

In general, the scheduling problem in an ARR system can be described as follows. Consider a list of one or more reservation requests  $R_1, R_2, \dots, R_q$ , where  $R_i = (c_i, s_i, h_i, p_i)$ . The first step is to establish an order in which

these requests are to be processed. The next step is to perform admission control and routing for each request on the list, according to the established order. Note that for immediate acknowledgment, the list always contains one request, and for delayed acknowledgment, the list corresponds to requests that are associated with a decision point.

For admission control, a request  $R_i$  is accepted if the network has sufficient capacity to support it, otherwise it is rejected (or blocked). Routing is then performed for the accepted request. This involves the determination of a tree that spans the participating parties. In our study, we use the minimum spanning tree (MST) heuristic [17] to construct the tree. This heuristic is relatively easy to implement. It achieves a tight performance ratio of  $2 - 2/k$  with overall worst-case time complexity  $O(kn^2)$ , where  $k$  is the number of parties, and  $n$  is the number of nodes in the network. In practice, the average performance of MST heuristic is much better than its worst bound.

The MST heuristic uses a link cost function for tree construction. Our cost function is based on an instantaneous link cost that is exponential in the utilization of that link. The instantaneous cost of a link  $e$  at time  $t$  is defined as:

$$w_e(t) = \delta + \mu^{\frac{b_e(t)}{N_e}} - 1, \quad \mu > 1 \quad (5)$$

where  $\mu$  is a chosen parameter,  $N_e$  is the total capacity of link  $e$ ,  $b_e(t)$  is the reserved capacity of link  $e$  at time  $t$ , and  $\delta$  is a small constant. The reason for the presence of  $\delta$  is that it prevents the cost of a link from dropping to zero when no reservation is made on that link. The cost function used by the MST heuristic is the average cost over the holding time of the request, say from  $t_0$  to  $t_1$ . This average cost at link  $e$  is given by

$$\bar{w}_e(t_0, t_1) = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} w_e(t) dt \quad (6)$$

The choice of  $\mu$  is discussed in [19]. It is found that as  $\mu$  increases, the load is more balanced at the expense of higher resource usage. It is also found that performance is not satisfactory when  $\mu$  is too large. In this investigation, we choose  $\mu = 2.0$ .

### 3.2 Heuristics

For delayed acknowledgment, we need a heuristic to order the requests that are to be scheduled at each decision point. Of interest are heuristics that are based on total resource usage. For our investigation, the total resource usage of request  $R_i$  is estimated by  $u_i$ , which is calculated as follows:

- Perform routing for  $R_i$ , assuming that all links in the network have full capacity, i.e., no advance reservation has been made. The result is a tree  $T_i$ .
- Calculate  $u_i$  as the product of total reserved bandwidth and holding time. That is  $u_i = (\sum_{e \in T_i} c_i) \cdot h_i$ .

Based on this estimate, we define the following heuristics:

- **Maximum estimated resource usage first (MAXEU)** - the requests are ordered in decreasing order of  $u_i$ 's.
- **Minimum estimated resource usage first (MINEU)** - the requests are ordered in increasing order of  $u_i$ 's.

Intuitively, the MINEU heuristic may be a good choice for minimizing the blocking probability since requests that require less resources are given higher priority. The MAXEU heuristic may be a good choice for load balancing because requests with large resource usage are scheduled first, and requests with small resource usage are scheduled later to fine tune the balance.

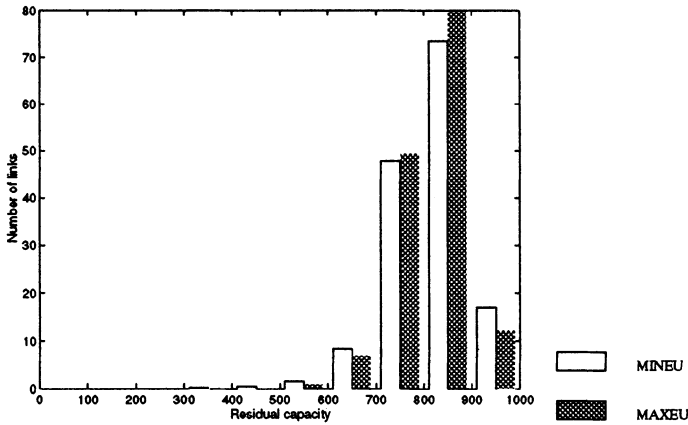
## 4 SIMULATION RESULTS

### 4.1 Preliminary observations

As a first step to understand the performance difference of MAXEU and MINEU for delayed acknowledgment, we use a simplified model which would allow us to obtain some initial insight on load balancing. In this simplified model, all requests have the same start time and same holding time. Requests are scheduled as a batch. The model parameters are as follows:

- Capacity of each link - 1000 units,
- Capacity requirement of each request - uniform over the interval  $[1, 75]$ ,
- Number of participating parties - binomial with mean 30,
- Number of requests in the batch - binomial with mean  $\bar{q}$ .

We run the simulation on 10 randomly generated networks and average the results. The values of  $\bar{q}$  are selected such that the blocking probabilities are negligible. Figure 3 shows the distributions of residual capacity in the network as a result of applying the MINEU and MAXEU heuristics. It is observed that MAXEU results in 29.9% fewer links in the low capacity region (with residual capacity  $< 700$  units) and 27.3% fewer links in the high capacity region (with residual capacity  $> 900$  units). The distributions show that the use of MAXEU can lead to a more balanced load.



**Figure 3** Simplified model: Distribution of residual capacity at  $\bar{q} = 15$

## 4.2 Comparison between immediate and delayed acknowledgment

We now evaluate the performance of the MAXEU and MINEU heuristics, when used in the delayed acknowledgment model. Of interest is their performance when compared to immediate acknowledgment (I-ACK).

In our delayed acknowledgment model, the notice interval is assumed to be uniformly distributed over the interval  $[Y + 1, T + Y]$ . This ensures that no requests require service before the first decision point. We choose  $T = 480$ ,  $Y = 80$  and  $P = 6$ . In this setting, all requests will be scheduled at the latest possible decision point since  $P \geq \lceil T/Y \rceil$ . The remaining model parameters are as follows:

- Capacity of each link - 1000 units,
- Capacity requirement of each request - uniform over the interval  $[1, 100]$ ,
- Number of participating parties - binomial with mean 40,
- Holding time - binomial with mean 20.

We study the performance of the scheduling heuristics as a function of the request arrival rate  $\lambda$ .

In our simulations, the parameters have been chosen to ensure small blocking probabilities. While a high blocking probability is acceptable in single-link models since it only causes the routing algorithm to choose alternative routes, it is hardly acceptable in a network model. Future networks are most likely to be designed with sufficient resources to ensure small blocking probabilities.

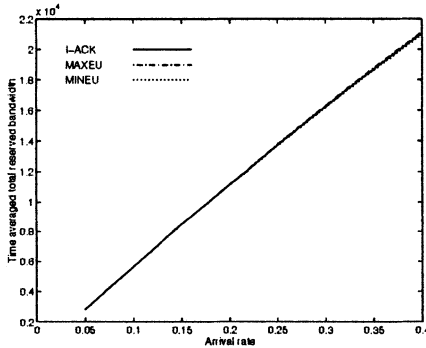
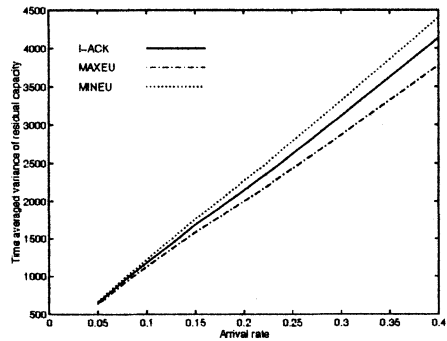
Simulation results for the blocking probabilities for selected values of  $\lambda$  are shown in Table 1. We observed no blocking for  $\lambda < 0.30$ . For larger values of



**Table 1** Blocking probability (%)

$\lambda$	I-ACK	MAXEU	MINEU
0.30	0.0010	0.0010	0.0010
0.35	0.0096	0.0154	0.0069
0.40	0.0125	0.0182	0.0089

$\lambda$  ( $\lambda = 0.35$  or  $0.40$ ) the MINEU and MAXEU heuristics yield the smallest and largest blocking probabilities, respectively. We conclude that for I-ACK, MINEU, or MAXEU, the blocking probability is negligible as long as the arrival rate does not exceed some specified value. Among the three algorithms, MINEU tends to yield the lowest blocking probability for a given arrival rate.

(a)  $\bar{C}$  as a function of  $\lambda$ (b)  $\bar{\sigma}^2$  as a function of  $\lambda$ 

**Figure 4** Performance comparison of immediate and delayed acknowledgment

Let  $\bar{C}$  and  $\bar{\sigma}^2$  be, respectively, the total reserved capacity and variance of residual capacity averaged over the entire simulation.  $\bar{C}$  and  $\bar{\sigma}^2$  are given by Equation (2) and (4) where  $t_0$  and  $t_1$  correspond to the start and end of simulation. In Figure 4a,  $\bar{C}$  is plotted against  $\lambda$ . We observe that there is little difference among the heuristics with respect to resource usage. Figure 4b shows the corresponding plot for  $\bar{\sigma}^2$ . We observe that MAXEU results in a smaller variance than I-ACK while MINEU results in a larger variance. The performance differences are noticeable, but not significant.

To gain further insight into resource usage and load balancing, we consider the total reserved capacity and variance of residual capacity averaged over

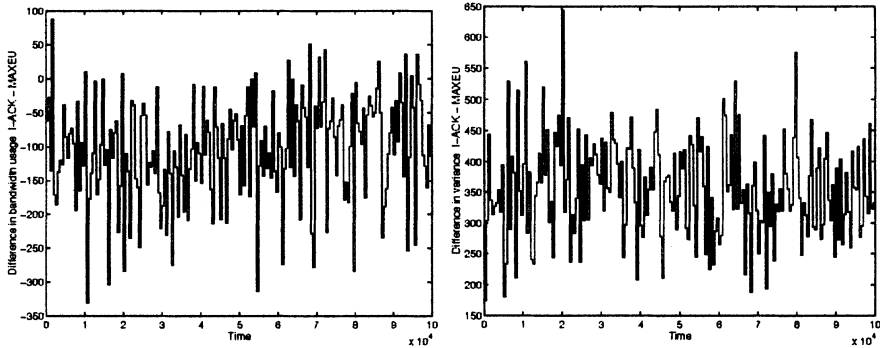
non-overlapping periods of 500 slots. The arrival rate  $\lambda = 0.40$ . Figure 5a shows the values of

$$\overline{C}_G(500i, 500(i+1)) \text{ for I-ACK} - \overline{C}_G(500i, 500(i+1)) \text{ for MAXEU}$$

and Figure 5b shows the values of

$$\overline{\sigma}_G^2(500i, 500(i+1)) \text{ for I-ACK} - \overline{\sigma}_G^2(500i, 500(i+1)) \text{ for MAXEU}$$

for  $0 \leq i \leq 200$ . We observed that for most of the time slots, MAXEU leads to slightly higher resource usage, but a smaller variance. Analogous results (not shown) are obtained for the difference between MINEU and I-ACK, namely, resource usage is similar, but MINEU has larger variance.



(a) Difference in  $\overline{C}_G(500i, 500(i+1))$

(b) Difference in  $\overline{\sigma}_G^2(500i, 500(i+1))$

**Figure 5** Performance difference between I-ACK and MAXEU at  $\lambda = 0.40$

## 5 CONCLUSION

In conclusion, delayed acknowledgment with the MAXEU or MINEU heuristic is similar to I-ACK in terms of resource usage. MINEU yields the lowest blocking probability but does not perform as well with respect to load balancing. On the other hand, MAXEU is better in terms of load balancing, but has the highest blocking probability. Unless load balancing is an important issue, it seems that I-ACK is the best alternative, especially when one considers the fact that it provides immediate response to the requester.

## REFERENCES

- [1] CCITT (1989) *Blue Book, Volume III, Fascicle III.7*, Integrated Service

- Digital Network (ISDN) - General Structure and Service Capabilities, Recommendations I.110-I.257, ITU, Geneva.
- [2] Handley, M., Schulzrine, H. and Schooler, E. (1997) SIP: session initiation protocol, Internet Draft, Internet Engineering Task Force.
  - [3] Harms, J.J. and Wong, J.W. (1995) Performance modeling of a channel reservation service. *Computer Networks and ISDN Systems*, **27**, 1487-97.
  - [4] Virtamo, J.T. (1992) A model of reservation systems. *IEEE Transactions on Communications*, **40**.
  - [5] Virtamo, J.T. and Aalto, S. (1991) Stochastic optimization of reservation systems. *Eur. J. Oper. Res.*, **51**, 327-37.
  - [6] Diaz Quinones, M.A. (1991) A simplified model of discrete-time reservation systems, in *Proc. 13th International Teletraffic Congress*, Copenhagen, Denmark, 647-52.
  - [7] Roberts, J.W. and Liao, K. (1986) *A queueing model of an advanced reservation system with blocked requests lost*, COST 214 Doc. 076.
  - [8] Liang, Y., Liao, K., Roberts, J.W. and Simonian, A. (1988) Queueing models for reserved set up telecommunications services, in *Proc. 12th. International Teletraffic Congress*, Torino, Italy, 4.4B.1.1-7.
  - [9] Reinhardt, W. (1994) Advance reservation of network resources for multimedia applications, in *Proc. 2nd International Workshop, IWACA '94*, Heidelberg, Germany.
  - [10] Degermark, M., Kohler, T., Pink, S. and Schelen, O. (1995) Advance reservations for predicted services, in *Proc. 5th International Workshop, NOSSDAV '95*, Durham, New Hampshire.
  - [11] Wolf, L.C., Delgrossi, L., Steinmetz, R., Schaller, S. and Wittig, H. (1995) Issues of reserving resources in advance, in *Proc. 5th International Workshop, NOSSDAV '95*, Durham, New Hampshire.
  - [12] Ferrari, D., Gupta, A. and Ventre, G. (1995) Distributed advance reservation of real-time connections, *Proc. 5th International Workshop, NOSSDAV '95*, Durham, New Hampshire.
  - [13] Winter, P. (1987) Steiner problem in networks: a survey, *Networks*, **17**, 129-67.
  - [14] Knuth, D.E. (1994) *The Stanford GraphBase : a platform for combinatorial computing*. Addison-Wesley, New York, 384-97.
  - [15] Charikar, M., Chekuri, C., Goel, A. and Guha, S. (1997) Approximation algorithms for directed Steiner problems, *Technical Report STAN-CS-TN-97-56*, Department of Computer Science, Stanford University.
  - [16] Cheung, T., Dai, Z. and Li, M. (1997) Approximating the Steiner problems on directed graphs, *Technical Report TR-97-1*, Department of Computer Science, City University of Hong Kong.
  - [17] Bharath-kumar, K. and Jaffe, J.M. (1983) Routing to Multiple Destinations in Computer Networks. *IEEE Transactions on Communications*, **Com-31**, 343-51.

- [18] Xu, C. (1998) Advance resource reservation networks, Master Thesis, Department of Computer Science, University of Waterloo.

## 6 ACKNOWLEDGEMENT

This work was supported by the Natural Sciences and Engineering Research Council of Canada.

## 7 BIOGRAPHY

Charlie Xu received the joint B.S. degree in computer science and mathematics from Simon Fraser University in 1996, and the M.Math degree in computer science from the University of Waterloo in 1998. His research interest is in the area of network resource management.

J.W. Wong received the B.S. degree in engineering, and the M.S. and Ph.D. degrees in computer science from the University of California at Los Angeles in 1970, 1971, and 1975, respectively. Since 1975, he has been with the University of Waterloo where he is currently a professor of computer science. From 1989 to 1994, he was Associate Provost, Computing and Information Systems. He was a visiting scientist at the IBM Zurich Research Laboratory from September 1981 to August 1982, from September 1988 to August 1989, and from September 1995 to August 1996. Dr. Wong was Editor for Wide Area Networks for the IEEE Transactions on Communications from 1989 to 1992, and served on the Editorial Board of Performance Evaluation from 1986 to 1993. He is currently on the Editorial Board of the IEEE/ACM Transactions on Networking. He was Technical Program Chair of IEEE INFOCOM '84 and of the 1994 International Conference on Computer Communications and Networks. His research interests include network resource management, distributed multimedia applications, and performance evaluation.

# Achieving 90% Throughput in A Flow-Oriented Input-Queued Switching Router System

*Geng-Sheng (G.S.) Kuo and Po-Chang Ko*

*National Central University*

*Department of Information Management*

*Information Technology Group*

*Chung-Li, Taiwan 32054 R.O.C.*

*Tel: +886 3 4263086; Fax: +886 3 4262309*

*e-mail: [gskuo@imrnet.mgt.ncu.edu.tw](mailto:gskuo@imrnet.mgt.ncu.edu.tw)*

## **Abstract**

Due to the head-of-line (HOL) blocking, the throughput limitation of an input-queued packet switching system with FIFO queues is  $(2 - \sqrt{2}) = 0.586$  (Karol, 1986) (Karol, 1987). Previous results show that for all independent and identical arrival processes, or packets, the throughput can possibly achieve 100%, if some suitable queueing policy and scheduling strategy are used (Anderson, 1993) (Karol, 1992) (McKeown, 1998). However, in the *real situation* all arrival processes are absolutely not independently and identically distributed, they are *flow-oriented*. The main purpose of this paper is to study the throughput in a flow-oriented input-queued switching router system utilizing the characteristics of a flow, and to design three scheduling strategies - SFROCF, SFRF, and LFRF. They can achieve 100% throughput for all independent and identical arrival processes and obtain a better throughput than that of McKeown et al.'s OCF (McKeown, 1993) (McKeown, 1998) in real flow-oriented input-queued switching router system on future *broadband* Internet. Among them, SFRF can achieve 90% throughput approximately, the best one, in the situation. In addition, the three strategies do not lead to the permanent starvation of a non-empty queue, because the weight value of a cell will increase if its waiting time increases. In closing, McKeown et al.'s OCF is a special case of our proposed scheduling strategies when the flow has one cell.

## **Keywords**

**Flow-oriented, switching router, throughput, OCF, SFROCF, SFRF, LFRF**

## 1 INTRODUCTION

Recently, the Internet is widely used and growing rapidly. The number of its hosts, www servers, and subnetworks has increased considerably (Gray, 1998) (Zakon, 1997). Much more multimedia-based *universal* information services will be developed on the Internet in the coming years. It implies that the traffic on future *broadband* Internet will be much busier than that of the current situation. Obviously, the router is going to be the most crucial bottleneck for future *global* broadband Internet. Therefore, it is necessary to improve the processing and performance of the router from an architectural viewpoint. The switching router is a new hot topic attracting attention in communications industry worldwide, which *integrates* switching and routing functions together to achieve better performance, e.g., IP switch, gigabit router, label switch, etc. (Newman, 1997).

However, it is well known that the maximum throughput of an input-queued switching router system is limited to  $(2 - \sqrt{2}) = 0.586$  derived by Karol et al. (Karol, 1986) (Karol, 1987), if the following five conditions are satisfied. 1. FIFO queues are used in an  $N \times N$  switching router, where  $N \rightarrow \infty$ . 2. Arriving packets are independently and identically distributed for each of  $N$  input lines during any one time slot. 3. The destined output line is uniformly distributed for each input line. 4. All arriving packets are of fixed and equal length, called cells. 5. Each input line has the same input probability distribution function (*p.d.f.*). Under these assumptions, the throughput limitation of an input-queued switching router system is due to the head-of-line (HOL) blocking.

Up to now, to the best of our understanding all studies and efforts on the throughput of the packet switching system are only concentrated on an input-queued switch, whose arriving cells are all independently and identically distributed. In the literature, many techniques, algorithms, and strategies have already been designed and proposed for the situation to reduce the HOL blocking and to increase its throughput (Anderson, 1993) (Chen, 1994) (Hluchyj, 1988) (Karol, 1992) (Obara, 1990). Both "*longest queue first*" (LQF) and "*oldest cell first*" (OCF) scheduling algorithms can achieve 100% throughput (McKeown, 1998). And, the *iSLIP* algorithm proposed in (McKeown, 1993) achieves 100% throughput by using some simulation method.

It is much more reasonable and realistic that the cells transmitted on real broadband Internet should be treated as many different sequences of cells. All cells in a *specific* sequence are coming from the same source node and destined for the same destination node, and are coined a *flow*. In other words, every cell arriving to a switching router can be treated as one cell belonging to some specific flow. All input flows are assumed to be independently and identically distributed. Every flow is labelled by one label number. Different cells belonging to the same flow are arriving to the same input line and destined for the same output line. The main purpose of this paper is to study the throughput in a flow-oriented input-queued switching router system, and to propose three scheduling strategies achieving 100% throughput in a cell-oriented input-queued switching router and obtaining

better throughput in a flow-oriented input-queued switching router by utilizing the above characteristics of a flow.

In Section 2, we briefly describe our switching router architecture, which is illustrated in Figure 1, and discuss the characteristics of a flow used in this paper. In Section 3, the scheduling strategies, LQF and OCF, are briefly mentioned. Although both two throughputs achieve 100% goal, they are *ideally* a maximum value. Some weaknesses need to be considered seriously on real Internet. At the same time, we propose our own scheduling strategies - "smallest flow rate and oldest cell first" (SFROCF), "smallest flow rate first" (SFRF), and "largest flow rate first" (LFRF). In Section 4, we specify our simulation model and method, and compare the simulation results derived from our scheduling strategies with the results obtained in (McKeown, 1993) and (McKeown, 1998). Finally, some conclusions are made that our proposed scheduling strategies are better than McKeown et al.'s OCF in a flow-oriented input-queued switching router system on future broadband Internet.

## 2 FLOW-ORIENTED INPUT-QUEUED SWITCHING ROUTER ARCHITECTURE

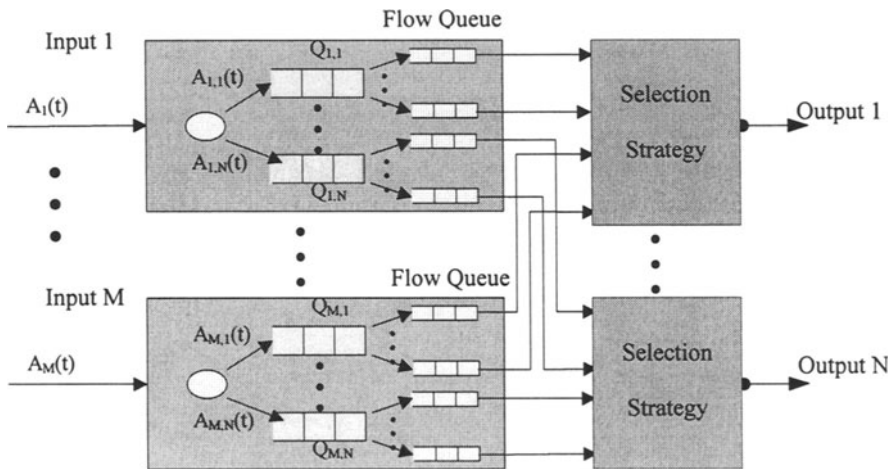


Figure 1. Every arriving cell is treated as one cell belonging to some specific flow in a flow-oriented input-queued switching router architecture.

There are  $M$  input lines and  $N$  output lines in our proposed flow-oriented input-queued switching router architecture illustrated in Figure 1. If one cell arrives from input line  $m$ , where  $1 \leq m \leq M$ , and is destined for output line  $n$ , where  $1 \leq n \leq N$ , this cell will be placed in queue  $Q_{m,n}$ . Because every arriving cell is treated as one cell

belonging to some specific flow, each cell entering  $Q_{m,n}$  is immediately transmitted to the corresponding flow queue according to its flow label number, FL. Let  $FQ_{i,n}(t)$  denote the  $i$ th flow queue destined for the  $n$ th output line during the  $t$ th time slot. From the viewpoint of output line, there are  $k$  flow queues,  $FQ_{1,n}(t)$ ,  $FQ_{2,n}(t)$ , ...,  $FQ_{k,n}(t)$ , serving the  $n$ th output line, where  $1 \leq n \leq N$ , during the  $t$ th time slot.

During the  $t$ th time slot, for every output line at most one cell is selected from one of these  $k$  flow queues,  $FQ_{i,n}(t)$  for some  $i$ , where  $1 \leq i \leq k$ , and transmitted through the architecture, if there is at least one cell existing in  $FQ_{1,n}(t)$  to  $FQ_{k,n}(t)$ . Let  $FT_{i,n}(t)$  denote the number of time slots spent on the cell arrival to the HOL of  $FQ_{i,n}(t)$  and at it during the  $t$ th time slot. In the proposed flow-oriented input-queued architecture, every flow has its own flow rate during the  $t$ th time slot,  $FR(t)$ . And,  $FT_{i,n}(t) = (1 / FR_{i,n}(t))$ . Let  $w_{i,n}(t)$  denote the *weight value* of the cell at the HOL of  $FQ_{i,n}(t)$  during the  $t$ th time slot, and  $W_{i,n}(t)$  be the *waiting time* of the cell at the HOL of  $FQ_{i,n}(t)$  during the  $t$ th time slot. In this paper, the weight value is defined by its scheduling strategy. And, the cell selected from these  $k$  flow queues and transmitted through this switching router for each output line is decided by the scheduling strategy accordingly.

### 3 SCHEDULING STRATEGIES – SFROCF, SFRF, AND LFRF

#### 3.1 LQF scheduling strategy

The LQF scheduling algorithm was proposed in (McKeown, 1993) and (McKeown, 1998). The algorithm can ideally achieve 100% throughput only under the input cell distribution is uniform. Otherwise, its throughput can not achieve the 100% goal. Furthermore, this makes the permanent starvation situation occur. This situation is demonstrated in (McKeown, 1998), which needs more attention to consider carefully.

#### 3.2 OCF scheduling strategy

The OCF scheduling algorithm was proposed in (McKeown, 1998). It considers the waiting time and assigns the weight value,  $w_{i,n}(t) = W_{i,n}(t)$ , to each cell at the HOL. If it has waited the longest time, the cell has the maximum weight value. During each time slot, OCF selects the cell with the maximum weight value to be transmitted. Therefore, no cell is starved for transmission indefinitely within a switching router system. Although it can overcome the permanent starvation of a non-empty queue, OCF still has one weakness behind. In a real physical situation, the traffic in a switching router system is flow-oriented. A flow is transmitted when one (VPI, VCI) pair exists, which is established between source node and destination node, in the switching router system. Two different flows may be established by two different nodes and have different flow rates, respectively.



Now, consider the following example. One  $2 \times 2$  switching router system is illustrated in Figure 2. Suppose two arriving flows are transmitted to output line 1. One flow arrives from input line 1 with  $FR=1$  and  $FL=1$ , and the other arrives from input line 2 with  $FR=0.5$  and  $FL=3$ . On the other hand, there is only one flow destined for output line 2, which also arrives from input line 2 with  $FR=0.5$  and  $FL=2$ . These three input flow orders and output cell orders by using OCF are shown in Figure 3. The number labelled in each time slot is the flow label number  $FL$ .

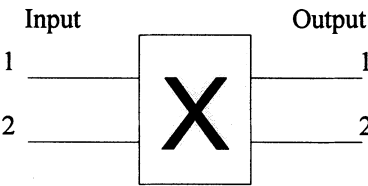


Figure 2. One  $2 \times 2$  switching router system.

In Figure 3, it is obvious that there are many idle time slots wasted in output line 2. If the cells of Flow 2 are selected prior to the cells belonging to Flow 1 or 3, a better throughput can be achieved than that of OCF. If  $FR$  of Flow 2 is much smaller than that of Flow 1 or 3, much more cells of Flow 1 or 3 can exist between two cells of Flow 2. Because cells belonging to the same flow arrive from the same input line and destined for the same output line, some cells, destined for different output lines, must belong to different flows. Therefore, if the cells of some flow with smaller  $FR$  can be transmitted with higher priority, a better throughput can be achieved. Based on this important idea pointed out here, the following three new scheduling strategies are constructed.

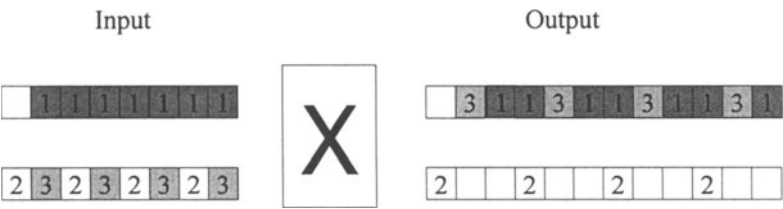


Figure 3. Three flows are transmitted through the  $2 \times 2$  switching router system using OCF scheduling strategy.

### 3.3 SFROCF scheduling strategy

The SFROCF is a new scheduling strategy proposed in this paper by us. In SFROCF, the weight value of the cell in HOL of a queue during the  $t$ th time slot is defined in terms of the following two components.

- $(1/FR(t))$ : If the cell belongs to a flow with a specific flow rate at the  $t$ th time slot,  $FR(t)$ , the *initial* weight value of this cell at the  $t$ th time slot is defined as  $(1/FR(t))$ .
- $W(t)$ : If the cell is destined for output line  $n$ , but is not selected to be transmitted during the  $t$ th time slot, its weight value will be increased 1 for the time slot.

The weight value  $w(t)$  of the cell is defined as  $(1/FR(t)) + W(t)$  by SFROCF. Its selection strategy is designed to select the cell having the maximum weight value. Although  $w(t)$  is assigned as  $(1/FR(t))$  initially, the cell, that has waited the longest time, will have the maximum weight value. It is very clear that no cell in a queue will be starved for transmission indefinitely.

Considering the example illustrated in Figures 2 and 3 again by using SFROCF, the input and output cell sequences are shown in Figure 4.

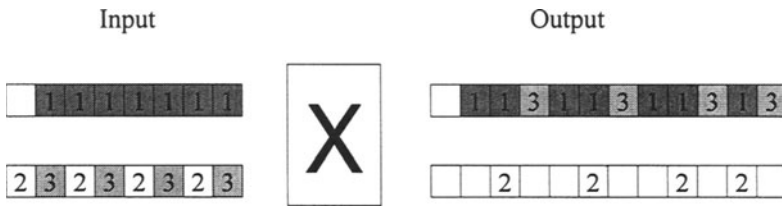


Figure 4. Three flows are transmitted through the 2x2 switching router system using SFROCF scheduling strategy.

Comparing Figures 3 and 4, it is obvious that the OCF spends 12 time slots to transmit these three flows, but SFROCF only spends 11 time slots. This situation is much clearer, if the difference between the maximum and minimum of all  $(1/FR(t))$  increases and the number of input lines increases.

### 3.4 SFRF scheduling strategy

Another new scheduling strategy SFRF is proposed in this paper by us too. Its weight value of the cell in HOL of  $FQ_{i,n}(t)$  during the  $t$ th time slot is defined in terms of the following two components.

- $(1/FR_{i,n}(t))$ : If the cell belongs to a flow with a specific flow rate at the  $t$ th time slot,  $FR_{i,n}(t)$ , the *initial* weight value of this cell at the  $t$ th time slot is defined as  $(1/FR_{i,n}(t))$ .
- $W_{i,n}(t)$ : If the cell is destined for output line  $n$ , but is not selected to be transmitted during the  $t$ th time slot, its weight value will be increased 1 for the time slot.

The weight value  $w_{i,n}(t)$  of the cell is defined as  $(1/FR_{i,n}(t)) + W_{i,n}(t)$  by SFROCF. Its selection strategy is designed to select the cell having the maximum weight value. The assigned weight value of the cell in HOL of  $FQ_{i,n}(t)$  during the  $t$ th time slot is the same as that of SFROCF. Its selection strategy is the same as that of SFROCF. The only difference between SFROCF and SFRF is the switching router architecture, where SFROCF uses the FIFO input-queued architecture, and the other uses the FIFO flow-oriented input-queued architecture shown in Figure 1. In SFRF, every arriving cell enters its corresponding flow queue. According to the assigned weight value of the cell in HOL of these flow queues, SFRF can select one cell with the maximum weight value and transmit it during each time slot.

Using SFRF to review the example illustrated in Figures 2 and 3 again, the input and output cell sequences are shown in Figure 5.

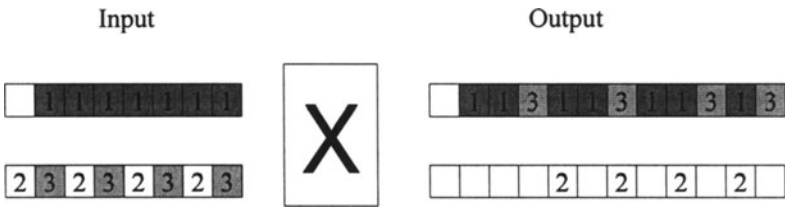


Figure 5. Three flows are transmitted through the 2x2 switching router system using SFRF scheduling strategy.

### 3.5 LFRF Scheduling Strategy

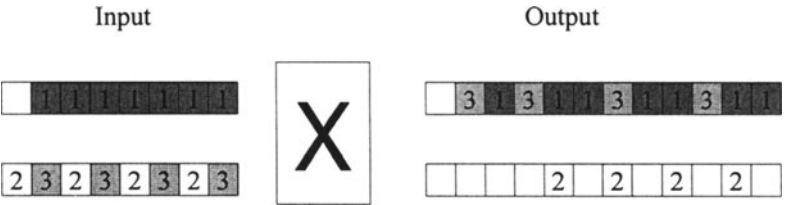


Figure 6. Three flows are transmitted through the 2x2 switching router system using LFRF scheduling strategy.

In LFRF, the assigned weight value of the cell in HOL of  $FQ_{i,n}(t)$  during the  $i$ th time slot is the same as that of SFRF. The switching router architecture used in this scheduling strategy is the same as that of SFRF too. The difference between SFRF and LFRF is that the FR of the flow, to which the cell selected by SFRF belongs, has the smallest value. On the other hand, if one cell is selected by LFRF, and the cell belongs to Flow  $x$ , the FR of Flow  $x$  has the largest value.

Similarly, using LFRF to review the example illustrated in Figures 2 and 3 again, the input and output cell sequences are shown in Figure 6.

4 SIMULATION RESULTS

In this section, we specify our simulation model and illustrate the simulation results in Figure 7 by using OCF, SFROCF, SFRF, and LFRF scheduling strategies, respectively. This simulation program was written in VB ver. 5.0 programming language and run in Microsoft Window 95 environment.

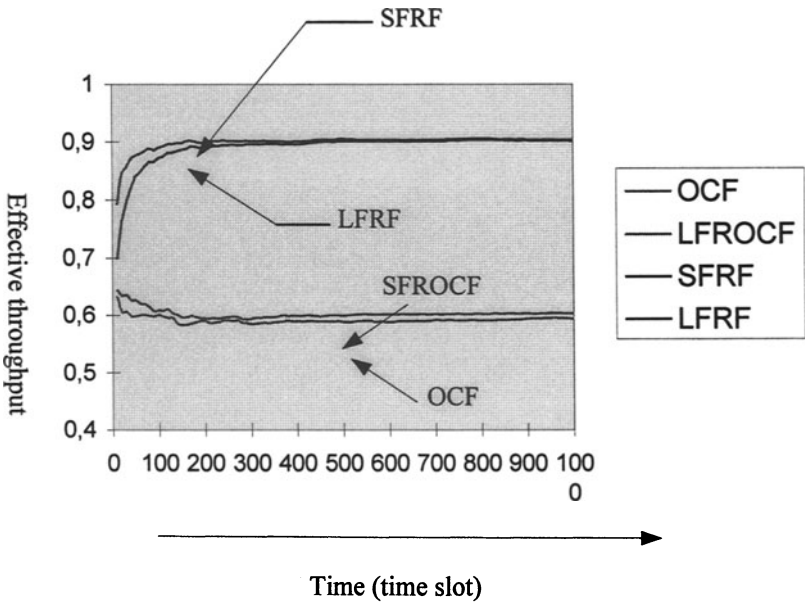


Figure 7. The simulation results using OCF, SFROCF, LFRF, and SFRF, respectively.

The arrival process used in this simulation model satisfies the following assumptions.

1. There is an arbitrary number of flow arrivals in each input line. All input flows are independently and identically distributed.
2. The (1/FR) of each flow is a random integer.
3. The cells belonging to the same flow have the same destination, that is, their output line numbers are the same.
4. The output line number, for which every flow is destined, is random.
5. Each cell in the HOL of a queue has an initial weight value according to the used scheduling strategy.
6. Considering the permanent saturation case, there is at least one cell arriving at each input line.
7. The calculation formula of effective throughput is the ratio of the total number of transmitted cells over the total number of arriving cells.
8. The switching router system used in this simulation model is 16x16.
9. The maximum number of traffic flows is 65535.
10. There may be one flow is ended and one new flow is generated during each time slot.
11. In this simulation model, the saturation case is considered.

If we neglect the transition state at the beginning of time axis ( $< 200$  time slots approximately), it is clear that both effective throughputs of LFRF and SFRF are 90%, and both effective throughputs of SFROCF and OCF are 60% approximately. Precisely speaking, the effective throughput of SFROCF is better than that of OCF. It is a fact that OCF can achieve 100% throughput under one important assumption that all arrival processes are independently and identically distributed. For the real situation, the flow-oriented approach is much more suitable. The cell currently transmitted on the Internet may have some relationship with the cells transmitted previously. This relationship is much more obvious in an input-queued switching router system. And, in this flow-oriented situation, our proposed three scheduling strategies can achieve better throughput than that of OCF. Among them, SFRF can achieve 90% throughput, the best one at this moment.

## 5 CONCLUSIONS

From the results in (McKeown, 1998), we have known that 100% throughput is achievable in an input-queued switching router system, if the specific assumptions are satisfied and a suitable queueing policy and scheduling strategy are used. The OCF proposed in (McKeown, 1998) can achieve the goal of 100% throughput under the assumptions. Among them, one very important assumption is that all arrival cells are independently and identically distributed. However, in the real situation all arrival cells are absolutely not independently and identically distributed, they are flow-oriented, especially in future broadband Internet. Therefore, it is more reasonable and realistic to consider the throughput in a real flow-oriented input-queued switching router system.

In this paper, we consider the flow as a unit for transmission in the real switching router system, and propose three new scheduling strategies - SFROCF, SFRF, and LFRF, in order to achieve a better throughput. Every arriving flow has a specific flow rate. If the flow rate is much smaller than that of other flows, much more cells of other flows can exist between two cells of the flow. And, different flows selected to be transmitted might obtain the maximum output. After careful and real simulation study, some meaningful results are obtained, which are very impressive to us. Our proposed three new scheduling strategies can achieve a better throughput than that of OCF. SFRF utilizes the characteristics of a flow, and achieves 90% throughput approximately, which is the best one at this moment. Furthermore, the permanent starvation situation does not occur in using SFROCF, LFRF or SFRF, because the weight value of the cell in HOL of a queue is composed of two components,  $(1/(\text{flow rate}))$  and *waiting time*. If one cell is not selected for transmission, its weight value will be increased. Eventually, the cell will have the maximum weight value and then be selected for transmission through the switching router system. In the future, it is our strong intention to improve the scheduling strategy for achieving 100% throughput in the real switching router system.

## 6 REFERENCES

- Anderson, T.E., Owicki, S.S., Saxe, J.B. and Thacker, C.P. (1993) High-Speed Switch Scheduling for Local-Area Networks. *ACM Trans. on Computer Systems*, **11**, no. 4, 319-52.
- Chen, M., Georganas, N.D. and Yang, O.W.W. (1994) A Fast Algorithm for Multi-Channel/Port Traffic Assignment. *Proc. of ICC '94*, 96-100.
- Gray, M.K. (1998) <http://www.mit.edu/people/mkgray>.
- Hluchyj, M.G. and Karol, M.J. (1988) Queueing in High-Performance Packet Switching. *IEEE J. Select. Areas Commun.*, **6**, no. 9, 1587-97.
- Karol, M.J., Eng, K.Y. and Obara, H. (1992) Improving the Performance of Input-Queued ATM Packet Switches. *Proc. of INFOCOM '92*, 110-5.
- Karol, M.J., Hluchyj, M.G. and Morgan, S.P. (1986) Input Versus Output Queueing on a Space-Division Packet Switch. *Proc. of GLOBECOM '86*, 659-65.
- Karol, M.J., Hluchyj, M.G. and Morgan, S.P. (1987) Input Versus Output Queueing on a Space-Division Packet Switch. *IEEE Trans. Commun.*, **COM-35**, no. 12, 1347-56.
- McKeown, N., Mekkittikul, A., Anantharam, V. and Walrand, J. (1998) Achieving 100% Throughput in an Input-Queued Switch (extended version). submitted to *IEEE Trans. Commun.*
- McKeown, N., Varaiya, P. and Walrand, J. (1993) Scheduling Cells in an Input-Queued Switch. *Electron. Lett.*, **29**, no. 25, 2174-5.
- Newman, P., Minshall, G., Lyon, T. and Huston, L. (1997) IP Switching and Gigabit Routers. *IEEE Commun. Mag.*, 64-9.

- Obara, H., Sasagawa, M. and Tokizawa, I. (1990) An ATM Cross-Connect System for Broadband Transport Networks Based on Virtual Path Concept. *Proc. of ICC '90*, 839-43.
- Zakon, R. (1997) Hobbes' Internet Timeline. *IETF RFC 2235*.

## 7 BIOGRAPHY

Geng-Sheng (G.S.) Kuo received his Ph.D. degree in systems engineering from Case Western Reserve University, Cleveland, Ohio, USA, in 1982. He then worked with R&D laboratories of the telecommunications industry in the United States, such as AT&T Bell Laboratories. In 1990, he came back to Taiwan and joined the Department of Information Management, Information Technology Group, at National Central University, where he is a professor. His current research interests include ATM-based broadband switching and networking technologies, and Internet service technologies. Currently, he is secretary of the IEEE Communications Society *Communications Switching Technical Committee*, editor for *Communications Architecture of IEEE Transactions on Communications*, a technical editor of *IEEE Communications Magazine*, etc.

Po-Chang Ko received his B.S. and M.S. degrees in electrical engineering from National Taiwan University, Taiwan, in 1989 and 1991 respectively. Since 1993, he has been an instructor at Van Nung Institute of Technology, where he teaches Computer Networks and Programming Languages. Currently, Mr. Ko is pursuing his Ph.D. degree in Information Management at National Central University, Taiwan. And, his research interests include design and implementation of switching router system, design of MAC for wireless LAN, and broadband Internet.

## ACKNOWLEDGMENT

This work was supported by the National Science Council of the Republic of China under Grant NSC 87-2416-H-008-006.

# Service Logic Mobility over Intelligent Broadband Networks

*Ch. Z. Patrikakis, S. E. Polykalas, I. S. Venieris*

*Electrical & Computer Engineering Department*

*National Technical University of Athens*

*9 Heroon Polytechniou Str., 157 73 Zographou, Athens, Greece*

*tel : +301 7722551, fax : +3017722534*

*e-mail : ivenieri@cc.ece.ntua.gr*

## **Abstract**

The concept of Intelligent Networks (IN) provides a convenient and future safe way for the rapid introduction of broadband multimedia services. The highly sophisticated design of multimedia services and the increasing demand of system resources creates the need for introducing intelligence on the end-systems. In the IN infrastructure this demand is balanced by the intelligence offered by the network itself. However, the task of deployment of new services or even porting of existing services between IN sub-networks is a tedious task that rarely makes use of the network's intelligence capabilities. The reason is the different implementations of the Service Logic (SL) in IN islands which makes it hard, if not impossible to re-use parts of the software over different IN platforms. This paper proposes ways that enable the remote use of IN capabilities by subscribers which are attached to another IN segment. In this attempt, interconnection issues are resolved using as tools the standard IN modules that is the Broadband Service Control Points (B-SCPs) and the Broadband Intelligent Peripherals (B-IPs). Deployment of these modules is based on the use of their service logic execution capabilities and in some cases on an enhancement of their traditional role. To provide the necessary platform and description tools for supporting service logic mobility, service design has been partially based on the concept of movable service logic scripts. Service distribution and service numbering schemes compliant to the network architectures are also provided. Finally, parallel to the description of the proposed models and architectures, practical examples are given.

## **Keywords**

Intelligent Networks, Multimedia Networking, Service Logic Mobility, Service Storage Server, IN scripts.

## **1 INTRODUCTION**



The idea of Intelligent Networks (IN) has lead into the development of network tools and infrastructure that allows for fast and efficient introduction of sophisticated services without the need of extended terminal upgrade. However, deployment of new services in new IN locations requires off-line updating of the networks in which services are introduced and porting of these services in a form that is suitable to run over the new network platforms. This is due to lack of a scheme that allows for introduction of services in a generic and platform independent manner over different IN networks.

On the other hand, a serious amount of traffic is imposed on the Intelligent Network in order to transfer information between the service logic execution points that is the Broadband Service Control Point (B-SCP) and the Broadband Intelligent Peripheral (B-IP) during service execution.

The solution to both aforementioned problems seems to be the adoption of a common service creation environment which is platform independent and can facilitate the transport and execution of service logic or service logic parts over different IN modules. This solution introduces the concept of movable service logic modules in the area of multimedia IN service creation and deployment. This concept can be applied to all IN modules directly involved with Service Logic and more specifically to :

1. The B-SCPs in order to exchange executable IN service parts or user related information such as user authentication - authorization information, offering literal service logic mobility over an IN Broadband Network infrastructure.
2. The B-IPs in order to exchange information related to resources used to support service logic mobility offered by the B-SCPs and to reduce the amount of information that needs to be duplicated or updated each time an IN service is introduced or modified.

To fully support the idea of service logic mobility over the network some basic principles need to be adopted. These can be summarized in the following :

- Use of service logic modules based on a platform independent implementation suitable for transport and execution over different IN elements.
- Development of the necessary environment capable of manipulating the transportation of service logic and the migration of service logic execution.
- Development of the necessary security mechanisms which will allow the safe transportation and testing of transported service modules on the host systems and which would guarantee faultless service logic execution.
- Development of a service distribution scheme which would minimize the exchanged information over the network in order to execute an IN service.

The paper is organized as follows. In Section 2 we present the general idea of service logic mobility as it will be used in the rest of this paper together with an example of a moving service. Section 3 gives details about the proposed network and system architectures. In Section 4 the numbering and addressing schemes deployed for supporting the presented architecture are described. Conclusions are summarized in Section 5.

## 2 SERVICE LOGIC MOBILITY IN BROADBAND IN

The main idea behind the concept of service logic mobility is to provide a scheme that allows the design of services that may be moved between IN modules, even ported to different IN systems. For this reason, the design of these services needs to be based on coherent independent modules which can be combined to produce integrated services. On the functional level this implies the design and implementation of independent re-usable service building modules which can be combined for the design of a service. The description of these modules could be scaled in different levels starting from the Global Functional Plane (ITU Q1203) or the Distributed Functional Plane (ITU Q1214) of the IN architecture. An example of such a service module combination on the Global Functional Level in order to form a service is the integration of an AUTHENTICATION, AUTHORIZATION, FIRST LEVEL SELECTION, PREVIEW, and REGISTRATION, DEREGISTRATION functional modules for providing a Video On Demand Service (Hussmann 1995). As one can see, several of these modules (i.e. AUTHENTICATION, AUTHORIZATION, REGISTRATION, DEREGISTRATION) could be re-used to construct other services. An example could be a Broadband Video Conference service that needs to make use of these modules. Throughout the paper the term 'script' is used in order to identify these modules. In other words a 'script' is an object - oriented self-contained entity that implements integrated and functionally independent parts of an IN service. Scripts can be transferred and executed either in the B-SCP or the B-IP. This is contrary to the narrowband IN approach where the service logic programs reside entirely inside the SCP. Obviously script development should be carried out with platform independent implementation object - oriented tools such as JAVA (Sun 1995) or Safe-TCL (Borenstein 1994).

By making use of the proposed ideas, an infrastructure on which portable services could be provided may be introduced. This infrastructure would allow the porting of a service between IN modules on two levels:

- The first is porting of services between IN modules of the same type such as B-SCPs (or even B-IPs). This allows for the movement of services or service building modules (i.e. AUTHORIZATION module) over the IN and also the use of a service or service parts over different IN implementations. Regarding the systems used to facilitate User Interaction such as B-IPs, the same logic would lead into customization of the user interface based on specific user information such as native language, disabled persons options, or even terminal specific information. This way, on one hand there is no need for duplication of services over different IN implementations, while on the other hand the customization of services and re-deployment over different environments based on specific user profiles becomes possible. Movement of services could be achieved by an B-SCP to B-SCP communication while customization of the user interface could be performed by a B-IP to B-IP communication. An example could be the deployment of a Video On Demand

service made possible over different networks based on the same User Front End and further more using a unique and global user profile meaning customized user - interface based on the native user's language.

- The second is porting of services between IN modules of different type such as B-SCPs and B-IPs. This could help the support of the same services even in environments that are built over different architectures. A possible example could be the deployment of a Video On Demand service which makes use of a B-IP even over an infrastructure that does not deploy a B-IP. This could be realized by reducing the functionality of the offered service related to the use of Special Resources (for example if a video preview option is offered to the user via the B-IP this could be omitted if a B-IP is not present). Another example is the transfer of service logic for the B-IP from central points implemented on B-SCPs.

Of course a coordinator of this service movement is needed in order to supervise the service migration procedure. Special service logic running on the B-SCPs could be used for this purpose. An inherent advantage of this idea of moving services is the easy introduction of new services or even the enhancement of existing ones. The idea of Service Storage Servers (SSSs) based on B-SCPs could be used for this purpose. This implies the implementation of several central B-SCP points, on which available services could be located and from which these services could be transmitted to the endpoint B-SCPs (client B-SCPs) interacting with the users. Therefore, introduction of a new service would rely on uploading of the service logic on the SSSs and notification of the client B-SCPs about the availability of this service. Caching mechanisms for downloading and temporary storage of services on the endpoint B-SCPs could apply reducing traffic based on service transportation requests.

Finally, several mechanisms for handling the requests for locating and downloading of services over the network should apply over the movement of services for ensuring faultless transmission. For this reason, specific service logic parts which could be integrated on the implementation platform of the end systems or embedded in the transmission protocols could be used.

Summing up the presented ideas and before we proceed with methods for supporting such an infrastructure, a simple example of a possible use of a service and the way the proposed infrastructure could handle the request and execution of this service will be presented. This is based on the architecture depicted in Figure 1.

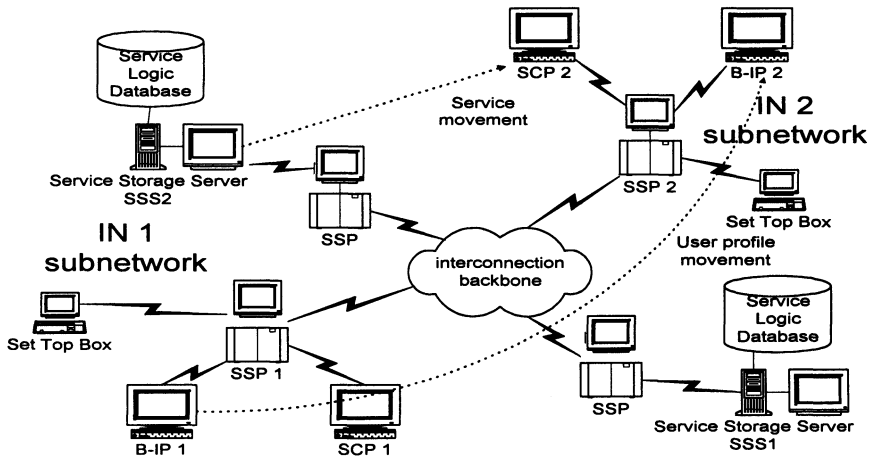


Figure 1 : An example of a scenario.

Let us assume that a user originated on Intelligent Network 1 is making use of several IN services based on a specific user profile. This profile could determine user settings such as native language, or customized registration Front Ends. This user is moving to IN2, while he wants to make use of a newly introduced service now available in all Service Storage Servers (i.e. SSS2). Let us assume that the user has the related software application installed on his terminal and is now requesting the use of this service. In section 3 we will see how this is possible even without using an IN infrastructure. Once the request for use of this service has been issued, the local Broadband Service Switching Point (B-SSP2) is requesting the execution of this service from the local B-SCP (B-SCP2). The B-SCP is responsible for locating the SSS2 containing the service logic and the user profile for customizing the service provision. The service logic is now downloaded from SSS2 to B-SCP2, while the preferences related to the user profile are transferred from the remote B-IP (B-IP1) to the local B-IP (B-IP2). Now the execution of the service is initiated on B-SCP2 while interaction with the user is performed over B-IP2 based on his native language. Furthermore, if we assume that registration to the service is required, this could be performed over the front end specified in the user profile.

### 3 THE PROPOSED ARCHITECTURE

#### 3.1 Requirements for service logic mobility

In order to support the idea of movable scripts, we must first provide a suitable architecture on which these scripts will operate. This architecture must comprise of:

- An implementation platform on which service logic could be executed independently of the underlying host platforms. The use of JAVA as such a platform seems to be an appropriate solution.

- A reliable and fast protocol for the transport of service or service modules over the network.
- A Service Storage Database and the related Database Management System for storing the service modules and handling service access requests. This could run on special service storage servers based on B-SCPs which could transmit the services to final destination B-SCPs or B-SSPs upon user request.
- A routing scheme for use during the movement of services. This routing scheme would have a dual use. The first is to provide a way of locating the SSS holding the service parts and the second is to locate the IN modules (B-SCPs and B-IPs) holding service user related information. This scheme could be based on the interaction between B-SCPs in order to schedule and route the transport of a service.
- A service logic implementation scheme for providing new services. As already mentioned such a suitable scheme is the use of IN scripts (Patrikakis 1997).

### 3.2 System architecture

Taking into account the study of the proposed architecture, one can see that a distributed architecture for providing a storage and distribution scheme, is the most suitable for supporting such a configuration. In this scheme we can distinguish several IN sub-networks consisting of client modules based on standard B-SCPs and B-IPs. These sub-networks are interconnected through an IN infrastructure serving a dual purpose. First to interconnect the IN sub-networks in order to facilitate the information exchange between these sub-nets and second to interconnect the clients with SSSs located in central points. This infrastructure is depicted in Figure 1 and is based on the deployment of the following components :

- The **SSS** which is built over an IN B-SCP and is used for storing the service logic information. In order to deploy a new IN service or service module, the supporting service logic needs to be uploaded on the server. A special SL daemon running the SSSs is used in order to notify all SSSs about the availability of this service. The server supervisors may decide about the policy of the servers related to the duplication of the service modules locally.
- The **client** which is also based on an IN module. The functionality of the client is the standard functionality of the host module (B-SCP, B-IP) enhanced with extra functionality in order to connect and download service modules from the SSSs or even from other clients. The client is not aware about the introduction of new services until a request for the use of such a service is issued. At this point, a communication with the closest SSSs takes place and the client is notified about the availability of the requested service.

Following, the architecture on which to base a server and a client system are presented.

#### *A. The Service Storage Server*

The server is located on a central point of the network infrastructure and may serve many IN sub-networks. It is based on the architecture depicted in Figure 2.

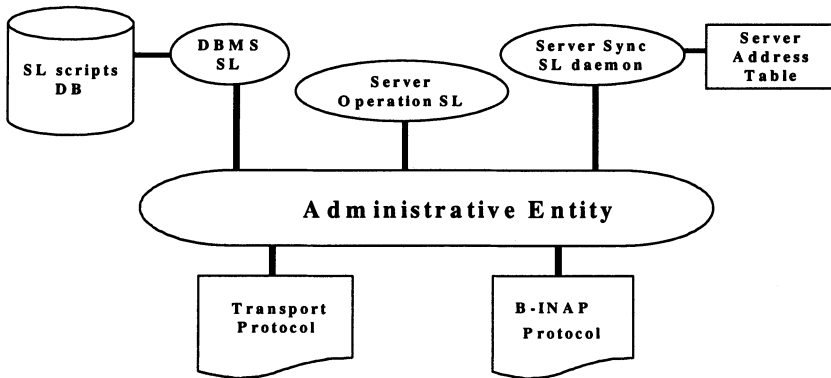


Figure 2 : Server Architecture.

The server consists of a core Administrative Entity which is acting as a manager of the server system. This entity handles all interactions and requests between the different modules of the server. The upper layer interface of this entity is used in order to interact with the service logic instances of the server while the lower interface is used for interfacing the protocols of the server. The service logic instances of the server are the following:

- The Server Operation SL which is the service logic responsible for performing system related operations on the server.
- The Database Management System service logic responsible for managing the database which stores the Service Logic scripts.
- The Server Sync Service Logic Daemon which is a service logic process running continuously on the server and which is used in order to keep the server always aware of the available services. For the latter, a server address table which is updated from the Server Sync SL daemon is used. This table holds the valid addresses of other SSSs on the network together with information about the location of services on these servers.

There are two protocols on the server:

- A transport protocol used for performing uploading and downloading of the service logic scripts from the clients,
- and a Broadband-IN Application Protocol stack (B-INAP) used for interfacing all IN modules (i.e. B-SCPs, B-SSPs).

The service logic scripts on the server comprise of two parts : The first is the main service logic script which runs on the B-SCP and implements the IN service and the other is a service logic script destined to run on the B-IP for supporting the B-SCP operation. The second script instance may be void if no B-IP deployment is necessary. Upon reception of a download request, the service logic scripts are sent to the destination B-SCP and B-IP. The entire procedure is manipulated by the B-SCP client which requested the download.

Let us see how the server operates by describing a scenario for the introduction of a new IN service. We will describe the process that takes place on the servers based on the architecture presented in Figure 2. First, the service is uploaded on a Service

Storage Server and a unique service identification number is assigned to the service. In section 4 we will present the scheme used for identifying services. To do so, the transport protocol is used for uploading the service to the SSS. During the upload of the service, the Database Management System (DBMS) SL is activated for storing the service logic on the SL scripts database. The service logic is implemented in the form of scripts.

Once the SL for the new service has been uploaded on the server, the Server Sync SL daemon is activated. The daemon sends a message using the B-INAP protocol to all servers listed on the Server Address Table. In case the service logic is not uploaded, the database of the SSS is updated with a link to the server that holds the service logic. Therefore, if a request is sent to an SSS for a service not available at the server's databases, it will be forwarded to the SSS holding this service logic. After this point the SSS is capable of transmitting the service logic to any B-SCP client upon request of this service. As one can see, this architecture enables a distributed storage of services, upon selective duplication of service logic scripts on the SSS databases.

### **B. The Client**

As it has already been mentioned, the client is a classical IN module based either on an B-SCP or a B-IP. A client may operate also as a server in case a request for downloading a user profile is received. This is necessary to reduce the amount of information maintained by the servers and to distribute it throughout the network on the B-SCPs and B-IP of the various sub-nets. Due to the dual form of the clients, we have two different architectures presented in Figure 3. The first is a reduced architecture based on an B-SCP and is depicted by modules drawn in solid line. The second is based on a B-IP and is depicted by all modules (drawn with solid and dotted lines).

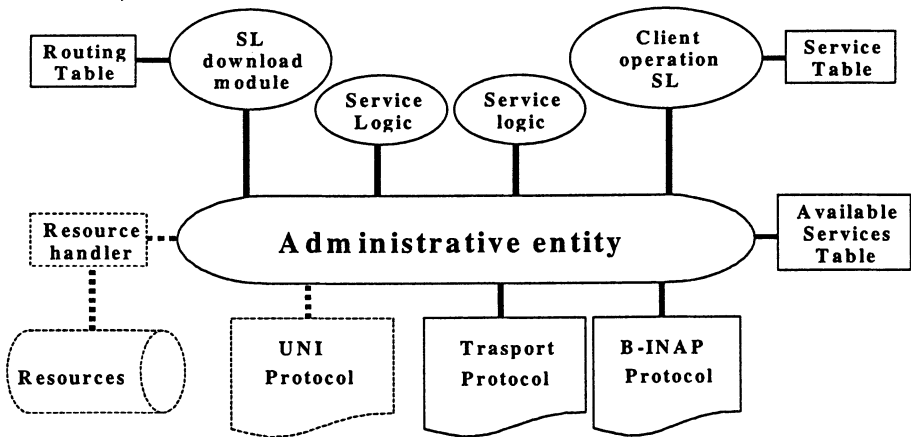


Figure 3 : Client architecture.

B-SCP based client.

This is the simplest form of client which is based on an B-SCP that may dynamically download and execute a service upon user request. It consists of the following components: A core Administrative Entity which manages all client modules and is responsible for the interaction of these modules. The implementation of the client Administrative Entity is similar to this of the server Administrative Entity. It uses a higher layer interface for the SL instances and a lower level interface for the protocols. The SL instances are the Client Operation SL which is the main SL instance performing system operations on the client and the IN service SLs which are the scripts implementing the services available at the client. The latter may be dynamically updated on the client upon user's request resulting to the availability different service logic scripts on the client at different times. An Available Services Table is kept in order to hold the identity of each supported service on the client. Finally, the SL download module which is used to locate a service on the network if this service is not supported from the client B-SCP at the time of request and to manage the download of the service.

The SL download module is also used to access a user profile located on another B-SCP client if such a profile is necessary for offering an IN service. Downloading of the service is performed in two steps: First the SL part running on the B-SCP is downloaded, and then if a B-IP SL part exists, the related SL is downloaded on the B-IP connected to the B-SCP by an B-SCP initiated call (optional). A routing table is used to locate the Service Storage Server or another client B-SCP. Regarding the protocols of the client we have a transport protocol used for downloading the service logic scripts, and a B-INAP protocol for interacting with the IN modules (i.e. B-SCPs, B-SSPs).

We will present the operation of an B-SCP client with an example based on the model illustrated in Figure 3. Let us suppose that a user located in a sub-network is making a request for a service. This request is forwarded to the closest client B-SCP and an instance of the service is checked to be found on the B-SCP. If the SL for supporting this service is not found on the B-SCP, then the SL download module takes over and the operation of locating the service is initiated. The service location procedure is executed based on the use of the routing table of the client B-SCP. The closest (or preferred) SSS address located in this routing table is contacted and a request for this service is sent. If the server contains the SL in its database then the service is downloaded on the B-SCP client. Else the server is transferring the request to another server which has this SL stored on its database. We must remind that if an SL running on a B-IP is necessary for the execution of the service, this is also downloaded to the B-IP corresponding to the B-SCP client. Finally, a check is performed and if the download of a user profile is necessary, the ID of the user that has requested the service is processed from the SL download module to determine the user's home network. The next step is to communicate with the B-SCP in the user's home network and to request the download of the user's profile. Once downloading of the SL for the requested service is finished, the SL instance is set active, the supported services table is updated, and a caching mechanism is initiated for this service. The service logic will remain on the client



B-SCP within a period in which no request for this service has not been issued. After this period the service logic may be discarded to make room for downloading another service logic script. This way, the service logic instances on the client B-SCPs are constantly updated upon user request.

#### *B-IP based client.*

This form of client is more complicated than the B-SCP based, and is depicted in Figure 3 by both solid and dashed lines. It uses the same architecture as the B-SCP based client, enhanced with a resource handler for manipulating the B-IP resources, the resources module and a UNI protocol stack (ATM 1994) for supporting the communication with the user. In this form of client, the IN service SL instances are SL programs used to support the service running on the B-SCP and their task is to perform the necessary interaction with the user upon B-SCP request. The SL download module is mainly used for handling any download of SL that is initiated from the related B-SCP once the user has requested a new service. This way, the B-IP does not deal directly with SL downloading since this is performed indirectly by the B-SCP. However, there are cases where a specific user profile is required to provide for a customized application front end for an IN service. The client B-IP offers this possibility by using the SL download module in order to contact the remote B-IP client and download this information based on the user profile which is available from the client B-SCP (see previous section). In order to do this a routing table similar to the one used in the B-SCP client is used. This operation implies that each user can be identified uniquely on a global basis as will be presented in section 4 below. The SL instances remain on the B-IP until the related B-SCP asks for withdrawal. The B-SCP caching mechanism is responsible for the definition of the time period for which the SL for a service is present on the B-IP.

Let us present the operation of a B-IP client following the example presented in the previous paragraph for the B-SCP client. We will make the assumption that the service requested by the user involves the use of a B-IP. In this case, after the B-SCP SL has been downloaded from the SSS, the B-IP SL is also downloaded and a check is performed to determine whether the service should be provided over a custom user profile or the user wants a customized front end. We assume that the user asks for a front end based on his native language. Such a front end is available only in his home network based on the home network B-IP resources. For this reason, first the user profile is accessed through the user's home network B-SCP client. Then the B-IP checks on the user profile and contacts the user's home network B-IP in order to download the necessary resources (native language interface). Now the service may be offered over this interface using the local B-IP.

#### ***C. User Terminals.***

The architecture presented in this paper was designed with the objective to keep service logic mobility transparent to the user's terminal. Only a "Navigation and Download application" is necessary for the terminals, enabling the download of software for deploying new services. This application can be based on widely used front ends that run over non IN networks. In this sense, the software for supporting a new service can be downloaded even from a non IN network (e.g. INTERNET)

by making use of this navigation and download application before the user has made a request for using a particular service. It is foreseen that special sites will exist where the users can find information on the newly introduced services and download the necessary software. Once the software is installed on their terminals services can be requested from every IN access point globally even through specific user profiles.

## 4 ADDRESSING AND NUMBERING SCHEMES

To support the architecture presented so far, a scheme for addressing the participating IN modules becomes necessary. In this section, possible realisations of such addressing schemes are given.

### 4.1 Service Storage Servers addressing

As mentioned, a Server Address Table is used in each SSS for storing the SSS addresses in the network. This table holds only a subset of all SSS addresses. Whenever a new service is introduced, notification to the SSS relies on the reproduction of notification messages from each server to all servers registered in its Server Address Table. This notification procedure is triggered on the SSS where the service was introduced. From this SSS, the first notification messages to the other SSSs on the network are transmitted. Upon reception of a notification message, each SSS forwards it to all SSS listed on its Server Address Table.

To avoid excessive traffic and useless multiplication of notification messages over the network, each SSS forwards the notification message to the other SSSs only once. Hence, only the original reception of a notification message triggers broadcasting on each SSS, while all other same notification messages received after that are ignored. The notification message encompasses the following information:

- Service Identification Number
- Necessary modules for deployment of the service (B-SCP, B-IP)
- Address of the SSS that holds the service logic for the new service
- A field indicating if the server is the initial server on which the service was uploaded or just a server holding a copy of the service.

Following this service distribution scheme, each SSS that reproduces the service by creating a local copy, inserts its own address on the third field of the notification message (SSS address that holds the service logic). Else the notification message would be reproduced and forwarded as it was received. In Figure 5, an example is presented. The gray filled circle indicates servers that will hold copies of the service while the white filled circle indicates servers that do not hold copies of the service. Finally, the solid line depicts the notifications that will be accepted while the dotted line notifications that will be ignored.

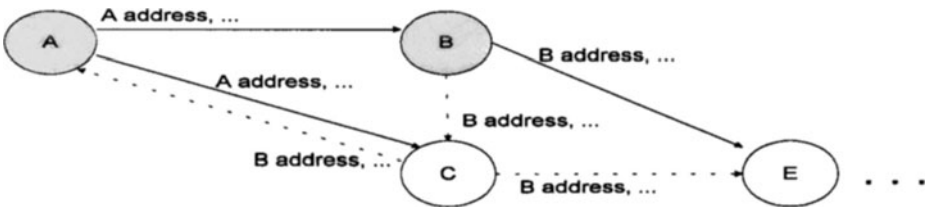


Figure 4 : Example of service notification messages.

Let us suppose that a new service is introduced on SSS A. The Server Address Table of A holds the addresses of B and C. Therefore a notification to these SSSs is issued with the SSS address field set to SSS A address. Server B makes a copy of the service and sends a notification to C and E by inserting its address on the server address field of the notification message. Notification to C is ignored since such a notification has already been received from A. SSS C does not create a copy of the service and issues a notification message to A and E by keeping the server address field of the notification message which now holds the address of A. Finally SSS E ignores the message of C since it has already been informed by B. After the exchange of messages for notification of a new service introduction has finished, the following information related to the availability of the new service will exist on the servers address table:

SSS	Type of service copy	Nearest SSS with service copy
A	original	A
B	copy	B
C	none	A
E	none	B

Table 1 : Server address table.

In the case of B-SCP clients, a routing table is used in order to access the closest SSS server. Whenever a service request is issued, the closest available SSS is contacted. To overcome failures in service requests in case an SSS is not operating, the addresses of several SSS are available in the routing table together with priority numbers. If a request to an SSS fails, a request to the SSS characterized by the next priority numbered is issued while the SSS which failed to respond is marked as unavailable. The next request starts with the SSS with the highest priority number among the available SSSs while a polling request to the unavailable SSSs is issued. Those who respond are marked as available and are restored to their priority numbers. If an SSS receives a request for a service for which it has no copy, this request is forwarded to the SSS related to this service in the server address table.

This scheme assures the distribution of services and links to servers across the network by referencing a server that is closer to the point where the service is requested. Alternatively to accepting only the first notification message, each SSS could have a certain time period in which all notifications received are accepted and processed for determining the closest SSS. The notification message originating from this SSS is accepted while all others are discarded.. A timer

initiated after the first notification has been received is used for setting the waiting time for the acceptance of notifications.

#### **4.2 Client addressing**

Apart from communicating with a server, a client may have to contact other clients in order to request user profile information or resources related to this profile. To do this, the client needs to know the user's home network. This problem is solved by introducing a global user ID number which identifies a user globally and is based on the following scheme :

Home Network Prefix	User ID
---------------------	---------

On the first part of the address the home network is identified while on the second a user ID is provided. Masking techniques could apply to both fields creating the idea of sub-networks and user groups. As one can see, this technique is similar to the one used in the IP addressing scheme (Socolofsky 1991) which is used for INTERNET accessing. Therefore, whenever a user requests a service, the local B-SCP decides about the user home network and treats any requests for user profile download accordingly.

#### **4.3 Service identification**

Taking into account the distribution scheme presented earlier for the introduction of new services one can see that the service identification scheme has to be dealt from the entity that uploads the service on the first SSS from which distribution and notification about the new service is initiated. Therefore, the service ID for each service or service module is assigned off-line. This is the easiest way in order to provide the same service ID request on the software necessary to be deployed on the terminals.

### **5 CONCLUSIONS**

This paper has investigated key architectural and design issues regarding network infrastructure related to the supporting of IN services over different IN implementations. The architectural schemes proposed in this paper provide the basis for an IN design that allows for quick and easy introduction of new IN services. The idea behind these architectures is the use of IN modules described in current IN recommendations, enhanced with service logic mobility support features. Service numbering and service distribution schemes for supporting service introduction under the presented architecture were also provided. Finally, implementation tools, as well as transport and signaling protocols capable of satisfying requirements posed by the proposed architecture were also proposed. The authors of this paper are currently engaged in the development of B-SCPs and B-IPs and have based the presented ideas on the experience gained in the context of ACTS INSIGNIA project.

### **6 ACKNOWLEDGEMENT**

This work was performed under the context of the EU ACTS Project MARINE (AC-340). The opinions appearing in this paper are those of the authors and not necessarily of the other members of the project consortium.

## 7 REFERENCES

- ATM User-Network Interface Specification V3.1 (1994) "af-uni-0010.002"
- Borenstein N. S. (1994) "E-mail with a mind of its own: The Safe-Tcl Language for Enabled Mail", ULPAA, Barcelona
- Hussmann H., Straten G.v.d., Theimer Th., Totzke J. (1995), "An IN-based Implementation of Interactive Video Service," in Proceedings of ICC'95, Seattle
- ITU Recommendations Q1203, "Intelligent Network Global Functional Plane architecture"
- ITU Recommendations Q1214, "Distributed Functional Plane for Intelligent Network Capability Set 1"
- Patrikakis Ch. Z., Venieris I.S., Protonotarios E. N., (1997) A modular Architecture for Broadband Multimedia Services Intelligent Peripherals", IS&N 97,15, Como
- Socolofsky T., Kale C., (1991) A TCP/IP Tutorial, RFC1180
- Sun Microsystems (1995) The Java Language Environment: A White Paper

## 9 BIOGRAPHY

**Charalampos Z. Patrikakis** was born in Athens, Greece, in 1970. He received the Dipl.-Ing. degree from the Electrical Engineering and Computer Science Department of the National Technical University of Athens (NTUA), Greece in 1993 and the Ph.D. degree from the National Technical University of Athens in 1997. Mr. Patrikakis is a research assistant in the Telecommunications Laboratory of NTUA performing research in the area of B-ISDN access networks, signaling, multimedia service design, and Intelligent Network technology. He has participated in several European Union projects and is currently involved in the INSIGNIA AC068 project. He is a member of IEEE and the Technical Chamber of Greece.

**Spyros E. Polykalas** was born in Kefalonia, Greece, in 1971. He received the Dipl.-Ing. Degree from the Electrical Engineering Department of the University of Patras, Greece in 1994. Since 1994 he is a Ph.D. candidate in the Electrical and Computer Engineering Department of the National Technical University of Athens (NTUA), Athens, Greece and research associate of the Telecommunications Laboratory. His research interests are in the area of signalling, Intelligent Networks, Personal Communications, performance evaluation and modeling. He has participated in several European Union ACTS projects and is currently involved in INSIGNIA AC068 and MARINE AC340 projects. He has received

several national awards for the entire five years of his undergraduate study. Mr. Polykalas is a member of IEEE and the Technical Chamber of Greece.

**Iakovos S. Venieris** received the Dipl. -Ing. degree from the University of Patras, Patras, Greece in 1988, and the Ph.D. degree from the National Technical University of Athens (NTUA), Athens, Greece, in 1990, all in electrical and computer engineering. He is currently an Assistant Professor in the Electrical and Computer Engineering Department of NTUA. His research interests are in the fields of B-ISDN, telecommunications software, internetworking, signalling, network management, modelling, performance evaluation and queueing theory. He has been exposed to standardisation body work and has participated in several European Union and national projects. Dr. Venieris is a member of IEEE and the Technical Chamber of Greece.